

**Computer-Aided Design Tools to Link Chemistry and Design in
Process Development**

by
Jonathan P. Knight

Submitted to the Department of Chemical Engineering in Partial
Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY
in Chemical Engineering

at the
Massachusetts Institute of Technology
February 1995

© 1994 Massachusetts Institute of Technology
All rights reserved

The author hereby grants to MIT permission to reproduce and to distribute publicly
paper and electronic copies of this thesis document in whole or in part.

Author

Department of Chemical Engineering
November 4, 1994

Certified by.....

Gregory J. McRae
Joseph R. Mares Professor
Thesis Supervisor

Accepted by.....

Robert E. Cohen, Professor of Chemical Engineering
Chairman, Committee for Graduate Students

MIT ARCHIVES
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

FEB 17 1995

Computer-Aided Design Tools to Link Chemistry and Design in Process
Development

by

JONATHAN P. KNIGHT

Submitted to the Department of Chemical Engineering on
November 4, 1994 in partial fulfillment of the Requirements
for the Degree of Doctor of Philosophy in Chemical
Engineering

ABSTRACT

Formal computer-aided design strategies are introduced to improve the identification, evaluation, and screening of process chemistries. A reaction path synthesis algorithm that combines retrosynthetic analysis and quantitative reaction prediction techniques has been developed for industrial aromatic chemistry to synthesize multiproduct, interconnected reaction paths. The algorithm has been verified through identification of both existing industrial routes as well as novel routes, yielding possible new raw materials, in-sourcing and out-sourcing of key intermediates, and recovery of valuable by-products. The reaction path network is used to derive a total plant synthesis problem and an efficient mixed-integer linear programming strategy to assist in the identification of promising reaction paths. Such an approach allows simultaneous consideration of reaction path alternatives and key process design variables. Computational results show that exploiting by-product recovery can lead to reduced waste streams and improved economics. A key challenge in quantifying potential benefits is the development of an integrated approach to obtain chemical information at appropriate levels of accuracy and cost. To address limitations in current group contribution methods, a strategy is created for synthesis of property models that exploits computed molecular surface areas and electronic density measures within a formal lattice potential. The approach leads to a considerable reduction in the number of fitted parameters. Furthermore, new model parameters are introduced that can be refined and transferred to multiple properties. In addition to extensive statistical evaluations and comparisons with existing methods, the new method has been demonstrated in a prototype property module to estimate ideal separation factors for aromatic isomer mixtures.

Thesis Supervisor: Dr. Gregory J. McRae

Title: Joseph R. Mares Professor of Chemical Engineering

Acknowledgments

I would like to express deep love and appreciation to my wife, Jennifer, for her love, support, and many sacrifices over the past four years.

Thanks to members of the research group for many useful discussions, particularly to Menner Tatang for his insights and algorithms in uncertainty analysis.

The research was supported by a National Science Foundation Graduate Fellowship Support Grant GER 92-53879 (1990–1993), a Cray Research Fellowship in Computational Chemistry (1993-1994), and a Kann Rasmussen Chlorine Project Grant (1993–1994). The research also made use of several grants in computer time for the Cray YMP/832 and the Cray C90 awarded by Cray Research at the Pittsburgh Supercomputing Center (1992–1994).

Table of Contents

List of Figures.....	13
List of Tables.....	19
1. Introduction	25
1.1. Thesis Statement.....	25
1.2. Research Introduction and Tenets	26
1.3. General Strategies and Techniques Employed	31
1.4. Research Tasks	34
1.4.1. Synthesis of Reaction Path Networks	34
1.4.2. Development of an Integrated Modeling Strategy	35
1.4.3. Integration of Reaction Path Selection and Process Design Decisions	36
1.5. Thesis Outline.....	37
1.6. Research Contributions	39
2. Synthesis of Reaction Path Networks for Systematic Identification and Analysis of Reaction Path Alternatives.....	40
2.1. Introduction	40
2.2. Overview	42
2.3. Industrial Insights	43
2.4. Introduction to Retrosynthetic Analysis.....	47
2.5. Reaction Path Synthesis Testbed	51
2.5.1. Local Retrosynthetic Analysis	51
2.5.2. Recursive Algorithm for Rigorous Synthesis of Chemical Reaction Paths	53
2.5.3. General Strategy for Implicit Enumeration.....	54
2.6. Reaction Prediction Techniques for Implicit Enumeration.....	59
2.6.1. Formal Representation of Competing Sites and Competing Reactions	62
2.6.2. Synthesis of Transform Networks.....	64
2.6.2.1. Monosubstitution Transform Networks: Homogeneous Nitration	64
2.6.2.2. Polysubstitution Transform Networks: Sandmeyer Reaction	67
2.6.2.3. General Polysubstitution Transform Networks: Chlorination.....	68
2.7. Computer Implementation	72
2.8. Example: Synthetic Musk Intermediate	74
2.9. Discussion.....	78
2.10. Conclusions.....	79
3. Integrated Frameworks for Reaction Path Synthesis and Process Development.....	80
3.1. Introduction	80
3.2. Overview	82

3.3. Review of Industrial Reaction Path Synthesis	82
3.3.1. Background	82
3.3.2. Original Chemical Engineering Approaches	84
3.3.2.1. Synthesis of Solvay-Type Clusters	84
3.3.2.2. Generalized Stoichiometry.....	86
3.3.2.3. Electron-Pushing Schemes	87
3.3.2.4. Retrosynthetic Analysis.....	93
3.4. Extensions of Reaction Path Synthesis of Relevance to Chemical Engineering	97
3.4.1. Chemical Modeling in Reaction Path Synthesis	97
3.4.2. DNA Synthesis Routes.....	99
3.5. Future Directions of Reaction Path Synthesis.....	102
3.5.1. Model Building: Stochastic Kinetics.....	102
3.5.2. Systematic Use of Chemical Databases: Industrial Case Study.....	107
3.6. Discussion.....	110
4. Quantum Chemical Calculations for the Estimation of Fundamental Molecular Parameters in Model Development.....	112
4.1. Introduction	112
4.2. Overview	114
4.3. Relevant Aspects of Quantum Chemistry.....	115
4.3.1. Schrödinger Wave Equation	115
4.3.2. Solution Structure.....	119
4.3.3. Solution of the Hartree-Fock Approximation.....	125
4.3.4. Choice of Basis Function Sets.....	131
4.4. Semi-Empirical Methods.....	132
4.5. Estimation of Properties and Their Modeling Applications.....	134
4.6. Comparison of Computational Methods	140
4.6.1. Molecular Geometry	141
4.6.2. Absolute Energies and Energies of Reaction.....	141
4.6.3. Ionization Energies.....	141
4.6.4. Dipole Moments	141
4.7. Discussion.....	142
4.8. Conclusions	143
5. Integrated Model Building Strategy Incorporating Molecular Calculations. Computational Tools.....	145
5.1. Introduction	145
5.2. Overview	147
5.3. Chemical Property Modeling	147
5.4. Incorporation of Molecular Calculations.....	149
5.5. New Strategy for Industrial Model Development.....	151
5.6. Computational Tools	153
5.7. Specification, Execution, and Archiving of Molecular Orbital Calculations.....	154
5.7.1. Quantum Calculations Databank.....	155
5.8. Model Building	156
5.8.1. Computation of Molecular Surface Area	156
5.9. Optimization Algorithm	161

5.9.1. Levenberg-Marquardt Algorithm.....	161
5.9.2. Solution of the Outer Problem: Linear Regression.....	163
5.10. Example: Estimation of Pure Component Properties.....	164
5.11. Discussion and Extensions.....	168
5.12. Conclusions.....	168
5.13. Supplement: Computer Implementation.....	169
5.13.1. Job Specification for Molecular Calculations.....	169
5.13.2. Specification of Model Parameters.....	174
6. Integrated Model Building Strategy Incorporating Molecular Calculations. Theory, Results, and Extensions.....	176
6.1. Introduction.....	176
6.2. Overview.....	178
6.3. Role of Interaction Potential in Property Relationships.....	179
6.4. Characterization of Intermolecular Potential.....	181
6.5. Pure Component Property Predictions.....	186
6.5.1. Model Results.....	187
6.5.2. Comparison With Method of Joback.....	190
6.5.3. Test-of-Fit for the Model.....	193
6.5.4. Statistical Significance of Electron Density Parameters.....	196
6.6. Transfer of Parameters to Other Properties.....	197
6.6.1. Octanol-Water Partition Coefficients of Nitroaromatics.....	198
6.6.2. Linear Solvation-Energy Relationship for the Menshutkin Reaction.....	201
6.7. Discussion.....	206
6.8. Conclusions.....	207
7. Integration of Reaction Path Selection with Process Design Decisions. A Total Plant Synthesis Approach.....	208
7.1. Introduction.....	208
7.2. General Strategy.....	211
7.3. Mathematical Programming Formulation.....	215
7.3.1. Flowsheet Material Balances.....	216
7.3.2. Feed Flow Design Equations.....	216
7.3.2.1. Efficient Formulation of the Design Equations.....	218
7.3.3. Material Conversion Design Equations.....	223
7.3.4. Unit Process Specifications.....	224
7.3.5. Mixed-Integer Linear Reformulation.....	226
7.3.6. Additional Mathematical Constraints.....	227
7.4. Model Extensions.....	227
7.4.1. Probabilistic Treatment of Uncertain Production Targets.....	228
7.5. Choice of Objective Functions.....	231
7.6. Computer Implementation.....	232
7.7. Conclusions.....	232
8. Verification of Reaction Path Synthesis Algorithm on a Large Case Study Involving a Spectrum of Related Products.....	233
8.1. Introduction.....	233
8.2. Overview.....	236
8.3. Chemical Conversions.....	236

8.3.1. Homogeneous Nitration (Monosubstitution Transform Network)	237
8.3.2. Molecular Chlorination (General Polysubstitution Transform Network)	238
8.3.3. Sandmeyer Reaction (Polysubstitution Transform Network).....	240
8.3.4. Béchamp Reduction (Monosubstitution Transform Network).....	241
8.3.5. Additional Reactions.....	242
8.4. Computational Strategy	243
8.4.1. Inference of New Reaction Rules	245
8.4.2. Partitioning of Chemical Reactions	248
8.4.3. Refinement of Reaction Prediction Parameters	248
8.5. Production Results	249
8.5.1. Excluding Hydro-Dediazonation	249
8.5.2. Incorporating Hydro-Dediazonation	255
8.6. Future Experiments and Areas for Problem Refinement.....	256
8.7. Discussion.....	258
8.8. Conclusions	260
8.9. Supplement: Calculation of Partial Rate Factors	260
8.9.1. Nitration.....	261
8.9.2. Chlorination.....	261
9. Integration of the Modeling Strategy with Process Scaleup Considerations: Separation of Reaction By-Products.....	262
9.1. Introduction	262
9.2. Overview	265
9.3. Property Estimation Module Prototype: Use of the Normal Boiling Point	265
9.3.1. Review of Computational Chemistry Tools.....	265
9.3.2. Components of a Property Estimation Module	269
9.4. Estimation of Relative Volatilities for Separation of Aromatic By-Products	270
9.5. Application to Column Design	273
9.6. Conclusions	275
10. Importance of Considering Multiple Production Routes in Preliminary Process Screening.....	277
10.1. Introduction.....	277
10.2. Overview.....	281
10.3. Problem Overview	282
10.3.1. Notation.....	282
10.3.2. Network Specifications	283
10.3.3. Unit Process Specifications	285
10.3.3.1. General	285
10.3.3.2. Assignment of Equipment.....	285
10.3.3.3. Equipment Design Estimates.....	287
10.4. Results from Deterministic Formulation.....	288
10.4.1. Computational Details.....	288
10.4.2. Study 1: Effect of Intermediate Prices.....	288
10.4.3. Study 2: Effect of Waste Stream Costs	292
10.5. Results for the Probabilistic Formulation	295

10.6. Discussion and Extensions.....	300
10.7. Conclusions.....	301
10.8. Supplement: Unit Process Specifications	301
10.8.1. Nitration	301
10.8.2. Chlorination	304
10.8.3. Béchamp Reduction	306
10.8.4. Sandmeyer Reaction for Nitrile Synthesis.....	307
10.9. Supplement: Additional Computational Details.....	310
10.9.1. Raw Material, Intermediate, and Product Values.....	310
10.9.2. Waste Stream Costs	311
10.10. Errata.....	312
11. Recommendations for Future Work	313
12. Conclusions.....	318
12.1. General Conclusions.....	318
12.2. Technical Conclusions.....	320
12.3. Concluding Summary	322
Appendix 1: Compound Databank.....	323
Appendix 2: Polaroid Case Study Report	332
Appendix 3: Introduction to Systems Engineering	339
A3.1. Overview	340
A3.2. Sequential Process Simulation	341
A3.3. Dynamic Simulation.....	342
A3.4. Equation-Based Process Optimization	343
A3.5. Integration of Process Subsystems.....	345
A3.6. Flexibility Analysis.....	347
A3.7. Modeling Uncertainty: Probabilistic Analysis.....	350
A3.8. Empirical Model Building	356
A3.9. Scheduling.....	360
Appendix 4: General Algebraic Modeling System (GAMS). Example Problem Specification and Results.....	363
A4.1. Deterministic Case Study: Nominal Intermediate Prices	364
Appendix 5: Code for Batch Distillation Module	415
A5.1. Computational Code for Batch Distillation Column Module	416
A5.2. Include File.....	432
Appendix 6: Code and Production Results for Reaction Path Network Algorithm	434
A6.1. Comments	434
A6.2. Computational Code.....	435
A6.2.1. Makefile	435
A6.2.2. Prune_Driver.f	436
A6.2.3. Ra_Driver.f	446

A6.2.4. Local_Ra.f.....	452
A6.2.5. Local_Ra_Phase1.f.....	455
A6.2.6. Local_Ra_Phase2.f.....	457
A6.2.7. Evaluate_Rxn_Path.f.....	463
A6.2.8. Get_Byprods.f.....	464
A6.2.9. Get_Rate_Data.f.....	466
A6.2.10. Get_Rxn_Data.f.....	474
A6.2.11. Update_Ast_List.f.....	475
A6.2.12. Update_Byprod_List.f.....	478
A6.2.13. Compare_Mol.f.....	479
A6.2.14. Compare_Mol2.f.....	481
A6.2.15. Params.f.....	483
A6.3. Production Runs.....	485
A6.3.1. Run #1: 3-Amino-4-Chlorobenzonitrile.....	485
A6.3.2. Run #2: 2-Amino-4-Chlorobenzonitrile.....	488
A6.3.3. Run #3: 2-Aminobenzonitrile.....	492
A6.3.4. Run #4: 2-Amino-6-Chlorobenzonitrile.....	500
A6.3.5. Run #5: 2-Chloro-4-Aminobenzonitrile.....	505
References.....	510

List of Figures

Figure 1.1. Road Map to the Integration of Process Generation, Evaluation, and Refinement Tools.....	33
Figure 2.1. Integrated Synthesis Routes Starting with 5-Nitroisophthalic Acid	44
Figure 2.2. Generation of Multiple Products Through Chlorination	45
Figure 2.3. Exploiting Choice of Reagent as a Basis for Producing a Useful By-Product in the Vilsmeier-Haack Formylation	45
Figure 2.4. Higher Isomeric Structures Defined by Hendrickson.....	46
Figure 2.5. Interconnections Between Isomer Forms in Industrial Aromatic Chemistry	47
Figure 2.6. Simple Illustration of Matching Between Functional Groups on the Target Molecule and Available Transforms to Yield a Proposed Retro-Reaction.....	48
Figure 2.7. Mapping of Functional Groups Onto the Aromatic Nucleus.....	49
Figure 2.8. Illustration of a Portion of the Retrosynthetic Network Generated for Musk Ambrette.	50
Figure 2.9. Graphical Representation of Target Molecule, Retrosynthetic Transformations, and Sub-Target Molecules	52
Figure 2.10. Generation of Precursors by Retro-Reduction of a Chloro-Phenyldiamine	52
Figure 2.11. Presence of Equivalent Sites on Target Molecule.....	52
Figure 2.12. Recursive Application of Local Retrosynthetic Analysis.....	53
Figure 2.13. Referencing of a Previously Explored Node to Avoid Duplication of Effort.....	54
Figure 2.14. Use of Evaluation Criteria to Eliminate Infeasible Retrosynthetic Conversions.....	55
Figure 2.15. Retrosynthetic Depiction of a Reaction Path Consisting of a Target Molecule, Retro-Transforms, and Sub-Target Molecules	56
Figure 2.16. Hierarchy of Constraints in the Generation and Evaluation of a Chemical Reaction Path.....	57
Figure 2.17. Illustration of How Pruning Can Eliminate Subpaths From Consideration	57
Figure 2.18. Retrosynthetic Nitration of m-Nitrotoluene that Generates Toluene as a Precursor.....	60
Figure 2.19. Application of a Sandmeyer Retro-Transform to Obtain Aniline from Benzonitrile.....	61
Figure 2.20. Product Distribution Associated with the Diazotization-Sandmeyer Reaction Sequence	61
Figure 2.21. Illustration of How the Set of Transformations Associated with a Chemical Reaction Can Produce Multiple Transformations of Functional Groups and Therefore Yield Multiple Products	62

Figure 2.22. Main Steps in the diazotization-Sandmeyer Reaction Sequence.....	63
Figure 2.23. Additional Aggregated Transforms Applicable to the Nitro Group Reduction to Account for Dinitro Reactants.....	63
Figure 2.24. Acetanilide Nitration Rates Predicted by the Additivity Principle.....	65
Figure 2.25. Main Steps in the Synthesis and Evaluation of Monosubstitution Transform Networks.....	66
Figure 2.26. Reduction of the Complex Reaction Mechanism to a Selectivity Analysis Based on Predominance of Ortho and Para Monosubstitution.....	67
Figure 2.27. Reaction Network for Batch Chlorination of Phenol.....	69
Figure 2.28. Ra_Driver: Overview of Retrosynthetic Analysis Algorithm Implementation	72
Figure 2.29. Local_Ra: Overview of Local Retrosynthetic Analysis and Reaction Prediction Implementation.....	73
Figure 2.30. Initial Fathoming of Target Molecule.....	75
Figure 2.31. Fathoming of 2-tert-Butyl-5-Methylanisole	76
Figure 2.32. Fathoming of 3-Methyl-4-Nitroanisole	77
Figure 2.33. Possible Routes to the Target Molecule.....	78
Figure 3.1. Simple Illustration of Retrosynthesis for Formylation by Formyl Fluoride.....	83
Figure 3.2. Graph Representation of a Solvay Cluster	85
Figure 3.3. Mechanistically Infeasible Reaction That Could Satisfy a Free Energy Screening Criterion.....	88
Figure 3.4. Interpretation of the Scheme for Improved Industrial Research and Development	88
Figure 3.5. Illustration of the Connectivity Matrix Representation.....	89
Figure 3.6. Illustration of the Definition of a Reactivity Matrix.....	90
Figure 3.7. Illustration of Different Reaction Matrices.....	92
Figure 3.8. Illustration of a Retrosynthetic Transformation.....	94
Figure 3.9. A Portion of the Candidate Transformations That Can be Generated for Methyl Cysanthamate.....	95
Figure 3.10. Outline of the Combined Retrosynthetic Analysis–Reaction Prediction Strategy	98
Figure 3.11. Typical Segment of Bihelical DNA.....	100
Figure 3.12. Possible Combinations of Raw Materials Used in the Synthesis of a DNA Segment	100
Figure 3.13. Illustration of the Procedure for Building Up a DNA Fragment	101
Figure 3.14. Chemical Modeling Formalism Being Developed by McDermott.....	103
Figure 3.15. Polyveratryl b-Guaiacyl Ether Compound.....	103
Figure 3.16. Model Compound Used to Obtain Intrinsic Kinetic Parameters	105
Figure 3.17. Depiction of an Interactive Retrosynthetic Strategy.....	108

Figure 3.18. Results of the CAS Search for Candidate Formylations	109
Figure 4.1. Depiction of Differential Overlap Integrals	134
Figure 4.2. Formal Reaction Products	136
Figure 4.3. Isodesmic Bond Separation Reaction	136
Figure 5.1. Model Building Strategy	148
Figure 5.2. Details of Model Building Strategy	154
Figure 5.3. Computational Architecture for the Specification, Execution, and Archiving of Molecular Orbital Calculations in the Cray C90 Environment	155
Figure 5.4. Program Design Language for Numerical Integration Algorithm	159
Figure 5.5. Comparison of Pseudo-Random (Multiplicative Generator) Points, HW Points, and a Uniform Grid	160
Figure 5.6. Comparison of Measured and Predicted Critical Temperatures for the Set of 42 Compounds	166
Figure 5.7. Comparison of Measured and Predicted Normal Boiling Points for the Set of 42 Compounds	167
Figure 5.8. Examples of Fragments	170
Figure 5.9. Sample Fragments Available in Job Specification Files	171
Figure 5.10. Sequence of Operations Performed on Fragments	172
Figure 6.1. Steps in Model Development	177
Figure 6.2. Property Estimation Strategy	177
Figure 6.3. Correlation of Critical Volume	182
Figure 6.4. Fit of the Normal Boiling Point	188
Figure 6.5. ANOVA to Determine Significance of Electron Density Parameters	192
Figure 6.6. Test-of-Fit of the General Model	194
Figure 6.7. Verification of Model	195
Figure 6.8. Depiction of the Menshutkin Reaction	202
Figure 6.9. Correlation of the Reaction Rate	204
Figure 7.1. Depiction of a Reaction Path Network	210
Figure 7.2. Partitioning of a Three Step Reaction Path Into Storage Units and Unit Processes	211
Figure 7.3. Introduction of Design Estimates to Simplify a General Process Model	211
Figure 7.4. Example Unit Process With a Variable Topology	212
Figure 7.5. Representation of the Mass Flows and Unit Processes	213
Figure 7.6. Arrangement of the Stream-types and Unit processes Into Stages	214
Figure 7.7. Superstructure of Net Material Flows Treated By the Unit Process Models	217

Figure 7.8. Illustration of How Non-integer Vertices Arise Due to Gap Between Upper Bound and True Variable Value	219
Figure 7.9. Comparison of Optimal and Robust Unit Process Configurations	232
Figure 8.1. Common Diazotization Sequences used in the Preparation of Specialty and Fine Aromatic Compounds	234
Figure 8.2. Series of Benzonitrile Dye Intermediates	235
Figure 8.3. Reaction Networks Yielding Multiple Substitution that Should be Considered as Part of the Reaction Prediction Problem for Chlorination	239
Figure 8.4. Benzotriazole Formation	240
Figure 8.5. Ammonolysis Reactions	242
Figure 8.6. Computational Strategy Employed	244
Figure 8.7. Existing Industrial Route Identified by Reaction Path Synthesis Algorithm	246
Figure 8.8. Alternative Route Involving a Chloroaniline Precursor	246
Figure 8.9. Candidate Route Involving a Polychlorinated Raw Material	247
Figure 8.10. Benzonitrile Dye Intermediate Series Not Including the Hydro-Dediazotation Transform	250
Figure 8.11. Existing Industrial Route Identified by Retrosynthetic Analysis Algorithm	251
Figure 8.12. Industrial Ammoxidation Route	251
Figure 8.13. New Reaction Paths of Potential Industrial Interest	253
Figure 8.14. Superstructure of Alternate Paths that Generates Operating and Inventory Alternatives	254
Figure 8.15. Highly Integrated Production Scheme that Exploits Hydro-Dediazotation Transforms	256
Figure 8.16. Example of the Potential For Yield Losses Due to Hydrolysis of the Nitrile	258
Figure 9.1. Model By-Product Recovery Problems Involving Mixtures of Aromatic Isomers	264
Figure 9.2. Computational Element for Estimation of the Normal Boiling Point	268
Figure 9.3. Prototype Assembly of New and Existing Computational Tools for Implementation in a Process Simulator Module	269
Figure 9.4. Estimation of Normal Boiling Point and Critical Temperature	270
Figure 9.5. Main Separations of Isomeric By-Products To Be Accomplished	271
Figure 9.6. Plot of the Pure Component Vapor Pressure and Separation Factor in the First Separation	272
Figure 9.7. Plot of the Pure Component Vapor Pressure and Separation Factor in the Second Separation	273
Figure 9.8. Block Representation of the Net Inputs and Outputs of the Computational Model	274

Figure 9.9. Theoretical Curves for Batch Distillation	275
Figure 10.1. New Reaction Paths of Potential Industrial Interest	280
Figure 10.2. General Set of Material and Energy Flows Entering and Leaving Unit Process Boundaries	282
Figure 10.3. Overview of the Design Specifications Used to Parameterize Equations in the Mathematical Programming Strategy	285
Figure 10.4. Set of Production Routes That Can be Generated	286
Figure 10.5. Reduced Network of Alternatives Based on the Elimination of Redundant Unit Processes	287
Figure 10.6. Optimal Unit Process for Nominal Intermediate Prices	290
Figure 10.7. Optimal Unit Process for Reduced Intermediate Prices	291
Figure 10.8. Optimal Unit Process for the Increased Waste Disposal Costs	293
Figure 10.9. Computational Results for the Chance-Constrained Programming Formulation	298
Figure 10.10. Further Computational Results for the Chance-Constrained Programming Formulation	299
Figure 10.11. Rough Sketch of a Unit Process for Homogeneous Nitration	302
Figure 10.12. Rough Sketch of the Béchamp Process for Nitrobenzonitriles	306
Figure 10.13. Rough Sketch of the Sandmeyer Unit Process	309
Figure 10.14. Identification Numbers for the Unit Processes for Assignment of Waste Stream Costs	311
Figure A2.1. Synthesis of DMB Based on Dehydrogenation of Dimethoxy Benzyl Alcohol	334
Figure A2.2. Chloromethylation Procedure Suggesting the Possibility of A High-Yield Approach to Hydroxymethylation	335
Figure A2.3. Chloromethylene Dibenzoate Condensation	335
Figure A2.4. Formylation by Formyl Fluoride	335
Figure A2.5. Fragmentation of a Secondary Alcohol	336
Figure A2.6. Manganese Mediated Formylation	336
Figure A2.7. Computational Results for the Ionization Potential	337
Figure A3.1. Simplified Illustration of Ammonia Process Flowsheet	346
Figure A3.2. Portion of the Optimized Heat Exchanger Network Developed for Ammonia Process	346
Figure A3.3. Simple Pump and Pipe Run for Flexibility Analysis Example	347
Figure A3.4. Determination of the Flexibility Index and the First Violated Constraint	349
Figure A3.5. Probabilistic Treatment of Input and Model Uncertainties to Determine Output Distribution	350

Figure A3.6. Robustness of the Net Present Value of a Process to Changes in an Input Parameter	351
Figure A3.7. Uncertainty in Catalyst Activity and Selectivity	352
Figure A3.8. Illustration of Tradeoff	354
Figure A3.9. Single Factor Binomial Lattice Used for Generation of Interest Rate Scenarios According to a Stochastic Differential Equation	356
Figure A3.10. Empirical Model Building Methods for Characterization of Steady-State and Transient Systems	357
Figure A3.11. Transient Response Surfaces for the Example Problem	359
Figure A3.12. First Three Orthogonal Eigenfunctions	359
Figure A3.13. Approximation of a New Observed Response	360
Figure A3.14. Graphical Depiction of Symmetric Traveling Salesman Problem for Four Cities	361
Figure A3.15. Different Types of Batch Scheduling and Design Problems	362
Figure A6.1. Correspondence Between Molecule Specification and Structure	485

List of Tables

Table 1.1. Some of the Major Levels and Chemistry-Related Factors Considered in Process Development.	26
Table 2.1. Partial Rate Factors for the Estimation of Chemical Reaction Rates	65
Table 2.2. Estimated Partial Rate Factors for Homogeneous Nitration	74
Table 2.3. Estimated Partial Rate Factors for Friedel-Crafts Methylation.....	74
Table 2.4. Estimated Partial Rate Factors for Friedel-Crafts Butylation	75
Table 3.1. Computed Parameters for Reaction Prediction	99
Table 3.2. Intrinsic Chemistry Studied by McDermott.....	104
Table 5.1. List of Main Compound Classes and Compound Databank Entries	156
Table 5.2. List of Compound Classes and Compound Databank Entries Used in Model Development.....	164
Table 5.3. List of Input Model Parameters and van der Waals Radii.....	165
Table 5.4. Summary of Results For Critical Volume, Critical Temperature, and Normal Boiling Point	167
Table 5.5. Objects Used in Job Definition.....	169
Table 5.6. Definition of <fgroup> Structure	169
Table 5.7. Definition of <molecule> Structure.....	171
Table 5.8. Specification of a Molecule Fragment	172
Table 5.9. Example of Job Specification File	173
Table 5.10. GAUSSIAN 92 Input File	174
Table 5.11. Format of Component Parameters Specification File.....	174
Table 5.12. An Excerpt From a Component Parameters Specification File	175
Table 6.1. Comparison of Computed Radii and van der Waals Radii.....	183
Table 6.2. List of Compound Classes and Compound Databank Entries Used in Model Development.....	186
Table 6.3. Model Parameters and Statistics Obtained for the Dataset.....	187
Table 6.4. List of Input Radii and Computed Charge Scaling Factors	189
Table 6.5. Model Accuracy for Different Compound Classes.....	190
Table 6.6. Breakdown of Model Fit and Joback's Predictions.....	191
Table 6.7. Set of 40 Compounds Removed to Perform Test-of-Fit.....	193
Table 6.8. Comparison of Model Statistics.....	195
Table 6.9. Breakdown of Model Fit into Individual Compound Classes	197
Table 6.10. Model Parameters and Statistics Obtained for the Prediction of the Octanol-Water Partition Coefficient	200

Table 6.11. Report of Experimentally Determined and Predicted Octanol-Water Partition Coefficients.....	200
Table 6.12. Model Parameters and Statistics Obtained for the Linear Solvation-Energy relationship.....	202
Table 6.13. Report of Experimentally Determined and Predicted Reaction Rates for the Menshutkin Reaction.....	203
Table 6.14. Model Parameters and Statistics for the Dataset used to Obtain Parameters for the Linear Solvation-Energy Relationship.....	204
Table 6.15. List of Input Model Parameters.....	205
Table 8.1. Nitration Partial Rate Factors.....	237
Table 8.2. Chlorination Partial Rate Factors.....	238
Table 8.3. The Set of Chemical Conversions and Their Underlying Chemical Transformations	245
Table 8.4. Nominal Values for Reaction Prediction Parameters	245
Table 8.5. Results from Variation in the Reaction Prediction Parameters.....	248
Table 8.6. Raw Data and Calculation of Partial Rate Factors for Nitration	261
Table 8.7. Raw Data and Calculation of Partial Rate Factors for Chlorination.....	261
Table 9.1. Results for the Binary Distillation.....	275
Table 10.1. Instances Considered for Chloronitroaniline Market Prices.....	289
Table 10.2. Optimized Configurations of Chemical Reaction Paths and Aggregate Tasks.....	289
Table 10.3. Optimized Configurations of Chemical Reaction Paths and Unit Processes under Different Waste Disposal Costs.....	294
Table 10.4. Candidate Total Production Objectives and Associated Probabilities	296
Table 10.5. Chance-Constrained Configurations of Chemical Reaction Paths.....	298
Table 10.6. Reaction Stoichiometry for Nitration	301
Table 10.7. Summary of Equipment and Utility Costing Equations for Nitration.....	303
Table 10.8. Reaction Stoichiometry for Chlorination in Acetic Acid.....	304
Table 10.9. Summary of Equipment and Utility Costing Equations for Chlorination.....	305
Table 10.10. Reaction Stoichiometry for the Béchamp Reduction	306
Table 10.11. Summary of Equipment and Utility Costing Equations for the Béchamp Reduction.....	306
Table 10.12. Reaction Stoichiometry for Sandmeyer Reaction.....	308
Table 10.13. Summary of Equipment and Utility Costing Equations for the Sandmeyer Reaction	309
Table 10.14. Commodity Values for Compounds in the Case Study.....	310
Table 10.15. Waste Stream Costs.....	312
Table A1.1. Compound Databank.....	324

Table A3.1. Two Widely Used Process Simulator Codes.....	342
Table A3.2. Chemical Engineering Applications of Mathematical Programming.....	344
Table A3.3. Codes and Recommendations for Linear Programming.	344
Table A3.4. Codes and Recommendations for Constrained Nonlinear Programming.	345
Table A3.5. Codes and Recommendations for Mixed-Integer Linear Programming.	345
Table A6.1. Substituent Identification Numbers in the Generated Compounds.	485

Chapter 1

Introduction

1.1. Thesis Statement

In the past chemical engineers have accepted the chemistry as a given, instead focusing their efforts on the design of process equipment. This approach is no longer enough if the chemical industry is to survive in the presence of tough economic competition and growing environmental concerns. A key premise of this thesis is that the chemical engineering profession must develop the technology to screen and ultimately select the chemistry it builds into processes, starting with the earliest stages of process development. Three key problems must be addressed. First, systematic representations of reaction path alternatives are needed. Second, design algorithms capable of analysis, screening, and optimization of problems with large information requirements and many constraints are essential. Third, the synthesis of chemical information should be treated formally through experimental design and model development and by optimizing the quality of information at each level of process development.

The enabling technologies are beginning to emerge. Computer-aided design now provides technical means for both systems design and information integration at all levels of process development, thus forming a basis for new approaches. Combinatorial design methods in systems engineering allow a more systematic representation and evaluation of large numbers of design alternatives. Computational chemistry techniques can significantly reduce the number of adjustable model parameters in property prediction models, thus reducing the cost of obtaining chemical information. In order to frame design alternatives generated by the chemistry, novel ways of representing process flowsheets are needed. Good representations must be relevant to the modern chemical industry and must establish a link between the underlying choice of chemistry and global design objectives. A basic conclusion is that, whereas unit operation-centered analysis dominates current thinking, additional types of process representations are needed to improve decisions at all levels of process development.

1.2. Research Introduction and Tenets

Chemistry is important to everybody, from the chemist who creates new molecules, to the engineer who designs chemical processes, to anyone who has ever needed a medicine, used a computer, or driven a car to work. The world is filled with decisions that require the expert and non-expert alike to evaluate and select among alternative chemicals and processes. For example, a doctor must use expertise in diagnosing ailments to determine which medicines are effective, safe, and reliable. By contrast many homeowners are novices when it comes to chemistry, yet they must still do enough research to select the right cleaning chemicals to clean and preserve many different surfaces. On a grand scale, chemical engineers engage in research and decision-making that is very similar to both examples when they develop new chemical processes. Table 1.1 summarizes some of the major issues in process development that are fundamentally dependent on the chemistry.

Table 1.1. Some of the Major Levels and Chemistry-Related Factors Considered in Process Development.

Screening of Basic Chemistry—Chemical Conversions, Raw Materials, Intermediates, Catalysts, and Product Molecules

- the ability to be incorporated in existing chemical plants
- presence of intermediates that are applicable to the development of multiple products
- innovations that lead to significant reductions in waste or improvements in safety

Evaluation and Scale-Up of the Small Number of Most Promising Chemistries

- measure and model chemical kinetics, toxicity of reaction by-products, impurity reactions
- identify project stopping variables (*e.g.*, poisons that cannot be removed from feed)
- compare chemistry-related tradeoffs (*e.g.*, better product selectivity versus faster catalyst decay rate)
- design the reactor configuration that best exploits the underlying chemistry

Refinement of Chemistry

- select/change the solvent, operating conditions, catalyst support, active ingredients
 - add chemistry to recover or recycle by-products
-

A basic message in Table 1.1 is that the treatment of chemistry is critically dependent on the evolution of process design issues. The identification of new processes begins with the analysis of basic chemistry developed in the laboratory. The relevant chemistry can be as limited as a single chemical conversions or can include an entirely new synthesis routes. From the set of alternatives, research and development priorities are allocated to the comparison and selection of the most strategically promising candidates. Once a small number of the most promising candidates are identified, it is necessary to refine and scale the chemistry to determine whether a process is feasible. The evolution in the understanding of the chemistry can be likened to a journey towards a distant goal—the key

features of the chemistry becomes increasingly defined as one follows a particular path towards specific process components.

This thesis is about the direct incorporation of process design tools into each level of process development to improve the identification, evaluation, and screening of alternative process chemistries. Because the chemistry and chemical plays such an important role at each level, better decisions among chemistry-related alternatives can improve the entire development strategy. Successful development of a chemical process(es) starts with the chemistry and requires considerable teamwork between many experts, including chemists, chemical engineers, project managers, and patent lawyers. These experts contend with multiple and often conflicting design objectives and constraints related to economics, competition, and the environment. Thus, any approach to decision-making must improve the communication of design objectives, constraints, and other information between experts in different disciplines who make the decisions. Improving the acquisition and dissemination of information alone, however, is not enough. In the presence of diverse sources of information and large numbers of design constraints, systems design tools are needed to assist in the analysis of alternatives and to optimize the final decision. Furthermore, such tools must play a vital role in identifying the most important experiments and model requirements for the efficient utilization of limited research resources.

Computer-aided design tools are an essential part of an approach and form the basis for this research. Very general computational tools such as spreadsheets can be adapted to millions of specific problems and allow a complex set of calculations to be performed easily for many different input specifications. State-of-the-art optimization algorithms allow design problems with thousands of variables and constraints to be solved, often in a period of only a few minutes. In short, the computer provides access to the only comprehensive set of building blocks for handling large amounts of information, representing design problems containing many variables and constraints, and exploiting efficient algorithms.

Although computer-aided design tools constitute distinct methods and specialties in their own right, a premise of this research that their application in process development (*e.g.*, measurement of properties, building of models, optimization of processes) is largely complementary and dependent on design decisions made at each level. A key limitation of current process design tools is that they are confined largely to unit operations. Process simulation and optimization methods, for example, are based on the integration of individual unit operations into process flowsheets for *given* chemistry. Furthermore, it is usually assumed that the models, data, and an organizational structure that can implement the results are available in an ideal form. As Table 1.1 shows, however, there are a number of key decisions which are encountered well before all such information is available. The development of appropriate tools to address these decisions must therefore exploit appropriate problem representations and also be flexible and undergo refinement as

new information becomes available. In fact, the cost and time expended in obtaining new chemical information and testing hypothesis in the laboratory significantly impact whether a computer tool can be implemented in practice and whether one can afford the answer.

This thesis addresses both the information and decision-making aspects in process development through the formulation and integration of relevant computer-aided design tools and problem representations. The mission of the thesis is to combine

- a modern view of industrial development;
- appropriate problem representations at different levels of development;
- state-of-the-art chemical information tools based on computational chemistry;
- efficient algorithms for large-scale analysis and optimization; and
- strategies for verification and improvement of methods.

The formation of specific technical objectives have been based on the following six tenets:

Tenet 1. The selection of a new chemical production route should be viewed as an integral part of a larger strategic plan that combines producing a spectrum of products with both economic and environment-related constraints and objectives.

Each chemical industry is composed of strategic sets of chemistries and process technologies that must be continually updated and improved to meet a wide range of market and regulatory changes. Such changes include newly discovered chemistry, evolving customer preferences, new patents, and new environment-related regulations, to name a few. Therefore, a plan should focus on answering the following questions:

- does the new production route make appropriate use of existing technology, raw materials, and intermediates; and
- what does the new route bring in terms of transferable technology, more generally applicable process chemistry, and safety- and environment-related improvements?

In short, an integrated approach to the selection of new production routes provides opportunities both to target desirable types of chemistry and process technology and to phase-out chemistries which are increasingly less desirable. Potential advantages of an integrated approach include:

- improving both process economics and the environment simultaneously;
- avoiding environment- and safety-related problems in the first place;
- systematically eliminating environment- and safety-related problems in existing processes, including generation of high levels of waste, inefficient or obsolete unit operations, and process operations which involve special safety hazards; and
- integrating design activities within a project as well as between different projects.

Tenet 2. Computer-aided tools are needed for virtually all design decisions encountered in real-world process development, not only for optimization of a final detailed design.

Tenet two has been discussed above.

Tenet 3. Evaluation of chemistry should incorporate process scale-up considerations and high-level design objectives in addition to lab-scale requirements.

Demand for greater product differentiation and increasingly specialized chemicals requires an ever-growing number of products, but in smaller volumes and with more cyclical demand patterns. In the past, environment-related objectives have traditionally been treated at the end of the design process (*e.g.*, through housekeeping practices, increased recovery of wastes, optimal control, and process design or retrofit), reduced waste generation and other process-related requirements must now be met early-on to compete effectively for research resources. As a result, industry increasingly recognizes the need to evaluate the large number of reaction alternatives developed in the laboratory to find strategic raw materials, intermediates, and chemical conversions. Furthermore, development of chemical reaction paths includes not only the attributes of the individual chemical conversions, but also the overall impact on in- and out-sourced intermediates, process integration, and waste streams. Tradeoffs in the choice of chemistry must be chosen to result in flexible and environmentally sound manufacturing strategies with intermediates being manufactured in enough volume to achieve an economical unit cost. Key technical challenges that must be addressed include

- problem representations and decision models that bridge the gap between the intrinsic attributes of the chemistry and high-level design objectives;
- improved techniques for the analysis of uncertainty to aid in the design of process scale-up experiments; and
- improved estimation techniques and model building frameworks for thermophysical and thermochemical properties.

Tenet 4. Optimizing the quality of information and allocation of limited research resources throughout process development are major practical motivations for new types of problem representations.

Process simulation tools generally require detailed models and chemical properties, the acquisition of which is a major task in process development. One of the crucial design decisions is, in fact, the decision to perform detailed measurements and modeling of a particular chemical or physical property. Identification and estimation of the key properties controlling process feasibility avoids unnecessary steps in development. For example, in a series of experiments designed to gain kinetic insights one might consider inserting a run

that contains realistic feed impurities. Demonstration that a feed impurity is detrimental to a catalyst has a definitive impact on the allocation of research resources towards feed cleanup, exploring higher catalyst loadings, or other measures; unless this key experiment is performed early-on, resources might be allocated to less important problems.

One of the requirements for optimizing the quality of information has already been discussed above, namely the capability to screen among a large number of alternatives in the presence of multiple design targets and constraints (as opposed to finding a single optimal design). In addition, problem representations are needed that are

- amenable to mathematical uncertainty analysis techniques to assess the significance of uncertainty in models on the selection among design alternatives; and
- allow one to differentiate which experiments can provide the most insight.

Tenet 5. The diversity and size of information requirements in process development motivate a much more integrated approach to property modeling in terms of the underlying model forms, design of experiments, and transferability of parameters.

Most any application of computer-aided design in industry requires development or refinement of models. Computer-aided design approaches do not replace experiments but rather change the set of experiments from direct system measurements to model building experiments. In this regard, current property modeling strategies face two key limitations. First, the number of parameters required to obtain accurate correlations can grow quite large. In group contribution methods, for example, differences between functional groups that can lead to different properties tend to be missed due to statistical averaging. Such averaging can be deleterious when developing good models for a particular class of compounds. Second, due to the need for good accuracy in models, the potential for exploiting cost-saving relationships between thermodynamic properties is often sacrificed in favor of empirical fits.

The past fifteen years have witnessed a revolution in new approaches to the acquisition of chemical information. Chemical databases have been created to provide rapid access to chemical preparations, thermophysical properties, environmental properties, *etc.* With greater computer power and better algorithms, computational chemistry techniques are being used more widely for the estimation of fundamental molecular parameters. Modeling tools incorporating these molecular parameters are being shown reduce the number of fitted parameters and allow for models that are transferable to different properties. While these advancements are occurring largely in the field of chemistry, their assimilation into general industrial research and development could significantly reduce the cost of modeling and experimentation. Key technical challenges include

- combining molecular orbital computations (and other types computational chemistry) with experimental data to obtain practical models that lead to improved estimates of thermophysical properties;
- integration of computational chemistry tools with other computer-aided design tools useful in process development (*e.g.*, process simulators); and
- demonstration that computational chemistry tools can significantly improve the ability to solve difficult and expensive design problems.

Tenet 6. New methods must combine technical scope and generality with the ability to solve industrial-strength problems.

Computer-aided design tools must be evaluated at two levels. First, the degree of generality of the tools should be established in terms of the underlying computational and algorithmic techniques. For example, property prediction models should be evaluated in terms of a general dataset of compounds in addition to specific compounds under consideration. Computational efficiency of numerical algorithms should be judged not only from the performance on individual problems, but more importantly from a general mathematical standpoint. Second, the tools should be combined and applied in unison to realistic industrial problems. Key metrics include the abilities to

- start with a problem specification and identify current industrial solutions;
- identify new or improved strategies; and
- facilitate new insights leading to a better understanding of the problem, suggesting specific experiments or further types of tools that can be developed.

1.3. General Strategies and Techniques Employed

Formal computer-aided design strategies are developed to address the six tenets listed above.

Reaction path synthesis is the enumeration and evaluation of reaction paths constructed from a specified set of basic chemistry (chemical conversions, raw materials, intermediates, and products). A particular type of reaction path synthesis is **retrosynthetic analysis**, which consists of the generation of reaction paths given the set of target products and desired chemical conversions; in effect, chemical conversions are applied in reverse to generate reaction paths that terminate with raw materials. In this research, a computational reaction path synthesis algorithm combining retrosynthetic analysis and quantitative reaction prediction techniques has been developed to synthesize a network of multiproduct, interconnected reaction paths, termed a **reaction path network** (chapters two, three, and eight). Construction of a reaction path network leads to a process representation termed a **chemical conversion-based representation**, in which each chemical conversion aggregates a portion of a complete process flowsheet around the main

reaction processes to create a set of aggregate material flows and energy balances. While reactants, products, and by-products always appear in the chemical conversion-based representation, solvents and other reagents usually are included only if they are not recycled internally. The chemical conversion-based representation facilitates analysis and comparison within the reaction path network.

Process synthesis is the integration of process elements to form a system capable of meeting specified targets and constraints. **Total plant synthesis** is a process synthesis problem which considers a multiplex of production capabilities and production routes to one or more products. Given a chemical conversion-based representation, the total synthesis problem requires considerable process screening. To assist in the identification of the most promising reaction paths in a process design sense, a form of the total plant synthesis problem is solved taking into account such factors as

- the alternate sources of raw materials, intermediates, and choices among products;
- in- and out-sourcing of intermediates; and
- choice of key separations.

A **unit process-based representation** of the chemical conversions is exploited that refines the chemical conversion into reaction, stabilization, and separations requirements. This representation merges the results from the chemical conversion level with estimates of yield losses, net separations requirements, and recycle requirements. Based on the unit process representation an efficient mixed-integer linear programming strategy is created that can solve large problems involving complex reaction path networks (chapters seven and ten).

A **model building strategy** is an approach combining computer model predictions, experimental data, model forms, and statistical fitting for the efficient and accurate estimation of chemical properties of interest. A key challenge in bridging the gap between the reaction path synthesis and process synthesis algorithms is the critical need for better model building strategies. To address limitations in current group contribution methods (as discussed above), a computational framework is created for synthesis of property models incorporating computed molecular surface area and electronic density measures obtained from molecular orbital calculations (chapters four, five, six, and nine). Such an approach leads to a considerable reduction in the number of parameters that must be fit. Furthermore, new model parameters are introduced that can undergo refinement and be transferred to multiple properties. In addition to extensive statistical evaluations and comparisons with existing methods, the new method can be employed in a prototype property module to estimate thermophysical properties for compounds where traditional group contribution methods fail.

A main goal of the research has been to verify the above methods in part by industrial examples. Figure 1.1 shows the formal interrelation of the tools discussed

above. The process development tools are categorized into three distinct but inter-related classes: generation tools, evaluation tools, and refinement tools. **Generation tools** are systematic representations of design alternatives at different stages of process development. Initially, a chemical conversion-based representation provides for the integration of alternative chemical conversions developed in the laboratory. Based on the selection of the most promising configurations of chemical reaction paths, the topological attributes of the reaction paths are considered to identify possible strategic raw materials and intermediates, as well as possible tradeoffs due to separations requirements, waste generation, and other factors. The results of such analysis defines the unit process-based representation, which is used for preliminary process screening to identify key areas for further laboratory development as well as use of modeling tools. Critical to the generation of design alternatives is their evaluation using **evaluation tools**. Reaction prediction tools based on linear free-energy relationships, kinetic modeling, reaction rules, treatment of competing reactions, and other relationships are employed to systematically eliminate unsatisfactory reaction paths from consideration. Chemical modeling tools are used to estimate needed thermophysical properties. In the analysis of integrated reaction paths, for example, estimation of thermophysical properties is critical to determine feasibility of separations of reaction by-products. Finally, **refinement tools** are needed to improve the generation and evaluation tools continually as new information is introduced. In addition to laboratory measurement of model compound information and direct system measurements, chemical databases and process modeling / simulation tools are essential.

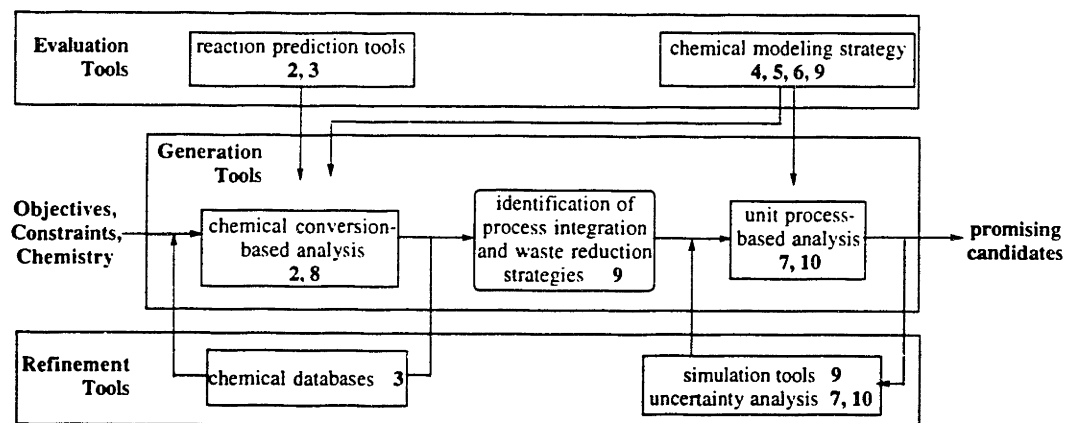


Figure 1.1. Road Map to the Integration of Process Generation, Evaluation, and Refinement Tools. Relevant chapters are given in boldface.

The complexity and diversity of modern industrial chemistry has motivated a model system of chemistry that is both broad in scope and that serves as a starting point for considering additional types of chemistry. Treated is a system of industrial aromatic chemistry that underlies a large number of processes and at multiple scales of production. Aromatic chemistry is useful as a model of organic chemistry for several reasons. First,

correlations have been developed in the literature to predict rates for several classes of reactions (although this is not available in practice, means we are not tied to the laboratory costs, *etc.*). Second, data from the dye, agrochemical, and drug industries are available for many of the relevant chemical processes. Finally, models of reactivity developed for aromatic chemistry are a good choice for extension to additional types of organic and even inorganic chemistry.

1.4. Research Tasks

1.4.1. Synthesis of Reaction Path Networks

A major goal is to gain a better understanding of how fundamental changes in raw materials, intermediates, and chemical conversions affect global process attributes, particularly when new chemical conversions are proposed to replace existing ones as part of a larger reaction path or reaction path network. While a number of process simulation and modeling tools are available to evaluate candidate process flowsheets, there is a considerable need to address process issues early-on in developing the chemical conversions themselves. One of the key attributes that cannot be determined by previous process synthesis or reaction path synthesis methods are structure-function attributes of the chemical conversions. Specifically, by adopting a chemical conversion-centered problem representation the strategic function of a chemical intermediate or chemical conversion can lead to important topological features that serve to integrate several reaction paths. Such paths are particularly relevant when considering common intermediates or conversions used in the manufacture of multiple products.

The systematic identification of topological features in the first place is a valuable aid to the strategic development of chemical intermediates and conversions, optimization of process boundaries, increased utilization of by-products to reduce waste, and a host of other objectives. Therefore, an algorithm is proposed for generating reaction path networks consisting of known chemical conversions with integrated topological features. The algorithm is based on a reaction path synthesis procedure which generates the complete set of product-yielding chemical reaction paths from a specified set of chemical transformations.

Several ingredients are critical to the success of the reaction path synthesis strategy. First, in order to deal with the interdependency between related reaction paths the synthesis strategy can handle multiple target molecules that might be found in multipurpose processes or that might use similar chemical conversions. Second, the synthesis strategy generates reactions containing reactant, product, and by-product molecules. Systematic generation of reactant molecules is essential to identification of common precursors. Similarly, identification of by-product molecules allows potential integration of the by-products to reduce waste streams. A two part reaction path synthesis strategy is used. Retrosynthesis is used to generate sequences of precursor molecules leading to economical raw materials.

Furthermore, as the precursors are generated they are "reacted" to identify major by-products using reaction prediction techniques. Efficient screening of the chemical reactions is based on an implicit enumeration strategy exploiting logical reaction rules, constraints on structural attributes of intermediates, and quantitative rate and selectivity targets. Reaction prediction techniques include the use of quantitative structure-reactivity models for systematic treatment of competing reactions.

Consideration of large-scale, multiproduct case studies is proposed as a basis for a realistic assessment of the algorithm developed in this work. Several metrics for evaluation of the algorithm are proposed:

- success in generating existing industrial routes as a set of sub-paths in the overall reaction path network;
- identification of novel, integrated routes involving novel raw materials or new applications of existing chemical conversions;
- identification of intermediates at several economic scales of production;
- generation of tradeoffs between the integration of reaction paths and process design issues such as by-product separability; and
- identification of concrete information requirements that result in testable hypotheses in the laboratory.

The degree to which such features can be identified should give a good indication of the richness of the proposed design space and the algorithm's capabilities. A key thrust of the research is to evaluate these metrics for a case study for a set of five commercial benzonitrile compounds.

1.4.2. Development of an Integrated Modeling Strategy

Preliminary process screening involves an extremely large number of design activities, including acquisition of chemical information, synthesis of separations systems, process simulation, dynamic simulation, scheduling, and a host of other design and operations-related activities. Major efforts to develop applicable design tools are currently underway in the chemical engineering community, particularly in the integration of process design and operations. The insights gained from a chemical conversion-centered analysis lead to several possible tools that can be integrated with other ongoing efforts.

One of the major challenges in incorporating process level considerations into the development of process chemistry are the large information requirements. Current modeling approaches such as group contribution methods require use of many different models for different properties, each involving determination of quite a few adjustable parameters to achieve good accuracy. In this research a strategy for chemical modeling is developed and utilized as an integral part of the chemical conversion-centered analysis. In

essence, the general strategy is to obtain transferable intermolecular potential parameters through a property model for a commonly available pure component property such as the normal boiling point. By exploiting a theoretical basis for the proposed property-energy relationships, the resulting intermolecular potential contributions are then be transferred to additional secondary properties of interest through fitting of models to model compound data. A basic advantage of the approach is that only a small number of contribution weighting parameters is required to correlate secondary properties. Parameters for the normal boiling point have been transferred to other pure component properties as well as partition coefficients and a linear solvation-energy relationship.

Implementation of the strategy involves considerable development of computational tools, including the integration of computational chemistry tools with efficient numerical integration and optimization codes. These features, combined with a systematic comparison of computational results with both data and other methods, suggests several basic questions to be addressed in the evaluation of the strategy:

- through the use of computationally efficient methods, how close can we come to a package that could be incorporated in a process simulator;
- how many fitted parameters are actually eliminated (in particular, can the use of binary parameters be avoided); and
- how transferable are the parameters to other problems (assuming a theoretical basis for the use of the parameters is available)?

1.4.3. Integration of Reaction Path Selection and Process Design Decisions

A basic message in this research is the need to consider process chemistry across multiple product lines so as to get at the multipurpose, strategic character of the raw materials, intermediates, and chemical conversions. This need motivates an approach involving the design of multiple chemical plants and taking into account common resource and technology requirements as well as synergisms between product lines. The chemical conversion analysis in fact motivates an approach to total plant synthesis based on a unit process-based decomposition. Such an approach can potentially use the reaction path network to characterize synergisms and tradeoffs in the choice of chemistry that ultimately affect the chemical process.

Of particular interest has been the representation of tradeoffs associated with the integration of chemical production routes and additional separations requirements. The network is recast as a partially aggregated process flowsheet consisting of appropriately defined storage elements and unit processes (*i.e.*, the aggregate set of unit operations associated with an individual chemical conversion). Scalable unit processes with static material conversion factors and variable topology can be derived from the full spectrum of process modeling tools, including process simulators. These results effectively

parameterize the separations alternatives, material conversions, and equipment design equations. Given a set of production targets and constraints, an efficient mixed-integer linear programming formulation of the combinatorial design problem has been derived. This formulation can be used in concert with a number of existing systems engineering tools, particularly process simulation and uncertainty analysis tools, to facilitate preliminary process screening.

This thesis has focused on development and verification of the unit process-centered analysis in three main areas:

- development of a computationally efficient mathematical programming formulation for problems that include a large number of alternate reaction paths, split fractions, and complex topological features;
- extension to deal with parametric uncertainty in material prices *via* chance-constrained programming;
- demonstration of the computational feasibility of the approach on a large-scale problem;
- gaining conceptual insight into general strategic and environmental considerations related to selection among raw materials, recovery of reaction by-products, and minimization of waste streams.

1.5. Thesis Outline

The thesis is organized into four main sections:

- the synthesis of reaction path networks (chapters two and three);
- the chemical modeling strategy (chapters four through six);
- integration of reaction path selection and process design decisions (chapter seven); and
- integrated application and verification of the proposed methods on an industrial case study (chapters eight through ten).

Following is a brief outline of each chapter.

Chapter two develops a retrosynthetic analysis algorithm for the systematic treatment of structure-function attributes of chemical conversions in a reaction path network.

Chapter three discusses implications of the algorithm for reaction path synthesis, particularly with respect to incorporation of new types of chemical information sources. Motivation for the chemical modeling strategy is presented. In addition, a general review of industrial reaction path synthesis is presented.

Chapter four introduces the quantum theory involved in the chemical modeling strategy.

Chapter five presents the computational tools, the amenable class of property models, and a preliminary demonstration of the chemical modeling strategy.

Chapter six gives a detailed accounting of computational experience with the chemical modeling strategy. Several issues are covered, including statistical significance of molecular parameters, head-to-head comparison with group contribution methods, and extension of pure component interaction parameters to the partition coefficient and a linear solvation-energy relationship.

Chapter seven develops the mathematical programming strategy for total plant synthesis and derives an extension to deal with uncertainty in production targets occurring on a time horizon that is short in comparison to project lifetime or campaign length. Two main theorems are proven.

Chapter eight defines a case study involving five benzonitriles and illustrates the application of the synthesis algorithm to create a reaction path network. Existing routes are generated as are novel routes based on the choice of a new raw material and recovery of reaction by-products. General insights gained from the results are the importance of reaction by-product recovery in reducing waste streams as well as the potential for simplifying chemical transformation to create entirely new classes of chemical intermediates for integrated reaction path development.

Chapter nine applies the chemical modeling strategy to the separation of reaction by-products present in an industrial reaction path network. The general problem of separating reaction by-products is more difficult than solvent separability due to the more diverse set of compounds involved. A case is considered where traditional group contribution methods attribute no distillation separability to a mixture of two aromatic isomers. Application of the new chemical modeling strategy estimates that the two isomers are separable with a relative volatility *ca.* 2. The thermophysical property estimates are incorporated in a batch distillation simulation module to estimate key design parameters.

Chapter ten develops a unit process representation of benzonitrile reaction path network. The most promising process configurations are identified as a basis for further research and development. In addition, the chance-constrained programming strategy for uncertainty in production targets is implemented. General process implications of reaction path networks for waste reduction and regulation in the chemical industry are discussed.

Chapter eleven presents future research directions.

Chapter twelve summarizes research conclusions.

1.6. Research Contributions

The main contributions of the research are as follows:

- elucidation of the strategic roles of chemical conversions and chemical intermediates in terms of structural attributes of a reaction path network, both in the abstract (chapters two and three) and on a specific industrial problem (chapter eight);
- development of a systems design algorithm for the synthesis of reaction path networks containing known chemical conversions (chapter two);
- development of a property modeling strategy for preliminary process screening that exploits molecular calculations (chapters four, five, six, and nine);
- development and illustration of a unit process-centered representation for total plant synthesis (chapters seven and ten); and
- extension of the total plant synthesis formulation to deal with certain forms of parametric uncertainty *via* chance-constrained programming (chapters seven and ten).

Chapter 2

Synthesis of Reaction Path Networks for Systematic Identification and Analysis of Reaction Path Alternatives

2.1. Introduction

The synthesis and screening of chemical reaction paths is a critical design activity that encompasses a growing number of economic and environment-related objectives. Whereas traditionally environment-related objectives have been treated at the end of the design process (*e.g.*, through housekeeping practices, increased recovery of wastes, optimal control, and process design or retrofit), reduced waste generation and other process-related requirements must be met early-on to compete effectively for research resources. Key stages in the development of chemistry including the following:

- (i) Discovery chemistry: new reactions that result in the formation of difficult-to-separate (*e.g.*, enantiomers) or environmentally undesirable by-products (*e.g.*, dioxins) are more likely to be eliminated in favor of stereoselective and environmentally benign routes.
- (ii) Chemical technology development: solvent, catalyst, and reagent properties are screened on the basis of environmental acceptability, safety considerations, and solvent recovery costs, to name a few.

- (iii) Integration of chemical conversions into new or existing processes: benefits of incorporating new chemical conversions, intermediates, and raw materials must be evaluated in terms of overall environmental impact, in- or out-sourcing requirements, and transferability of the technology to interdependent product lines (*e.g.*, in a multipurpose process), to name a few.

While a number of applicable tools, including retrosynthetic analysis, chemical information databases, and thermophysical models, have been developed to assist in stages (i) and (ii), there is a considerable need for systematic approaches to stage (iii) *early* in the development of reaction path alternatives. The scope of design issues encountered in stage (iii) is quite broad since it involves fundamental choices among raw materials, intermediates, and chemical conversions that must be evaluated from a total process perspective. First, there is a need to assess the overall impact of a new chemical conversion on total process flows such as energy requirements and waste stream generation. For example, replacement of an existing chemical conversion by a more "environmentally benign" conversion could reduce the level of energy integration, inadvertently resulting in increased utilities costs. Second, evaluation of the demand for new raw materials and intermediates is essential in order to determine whether they can achieve an economic scale of production. Such an evaluation can be critical both in the development of individual chemical processes (*e.g.*, multipurpose batch processes) and in the development of intermediates that can be used by many processes. Third, implicit costs related to changes in process flexibility and future environmental regulations must be carefully assessed. Of particular importance is the uncertainty in operating costs and inventory policies associated with in-sourcing and out-sourcing of intermediates.

An important aspect of these issues that has not previously been considered is the relationship between the strategic function of a chemical intermediate or chemical conversion and the resulting impact on the topology of an overall reaction path network—*i.e.*, an integrated configuration of strategic raw materials, chemical conversions, and products required to achieve an economic scale of production. Systematic identification of topological features in the first place could be a valuable aid to the strategic development of chemical intermediates and conversions, optimization of process boundaries, increased utilization of by-products to reduce waste, and a host of other objectives.

An algorithm is proposed for generating reaction path networks consisting of known chemical conversions with integrated topological features. The algorithm is based on a reaction path synthesis procedure which generates the complete set of product-yielding chemical reaction paths from a specified set of chemical transformations (an introduction to reaction path synthesis is given in section four of this chapter; an extended discussion is given in chapter three). In order to deal with the interdependency between related reaction paths, the synthesis strategy can handle multiple target molecules that might be found in multipurpose processes or that might use similar chemical conversions.

Reaction path synthesis techniques (Agnihotree *et al.*, 1979; Govind *et al.*, 1980; Nishida *et al.*, 1981) have traditionally been limited by the need to evaluate a large number of generated reaction paths. Experimental screening of hundreds or thousands of alternate reaction paths is impractical in most instances, especially since many of the proposed routes are likely to either be manifestly infeasible or to require extensive modification in the laboratory. To address these limitations, the algorithm readily exploits chemical information sources such as chemical databases, reaction prediction models, and reaction rules to prune away infeasible reaction paths and to facilitate comparison of generated reaction path alternatives. In particular, reaction prediction techniques using these tools facilitate preliminary analysis of reaction feasibility, incorporation of competing reactions, and estimation of product fractions. Additionally, the approach can potentially be useful in synthesis of new reaction paths for related products. In particular, it may be possible to start with limited data and models and to successively refine a set of candidate pathways to a smaller set worthy of further development. Since the approach results in testable predictions, it may also aid experimental design.

Aromatic chemistry is chosen as a testbed for the algorithm for several reasons. First, a theory of quantitative structure-reactivity models is already available for a number of its reaction classes (Exner, 1988). Second, data from the dye, agrochemical, and drug industries are available for many of the relevant chemical processes. Finally, the resulting models for reactivity and physical properties are applicable to additional types of organic and even inorganic chemistry.

This chapter is organized as follows. First, industrial examples are presented to elucidate structure-function attributes of chemical reaction paths and conditions under which they are desirable. Based on a general representation of such topological features, a retrosynthetic analysis algorithm is developed that is well-suited to the generation of integrated reaction path networks. Presented are the analysis of individual conversions, the representation of competing reactions, the reaction pruning strategy, and the computational implementation. Finally, the algorithm is illustrated with a commercial synthetic musk.

2.2. Overview

The strategic function of a chemical intermediate or chemical conversion can lead to important topological features that serve to integrate several reaction paths, particularly when considering common intermediates or conversions used in the manufacture of multiple products. These features are generalized through the concept of a reaction path network. A retrosynthetic analysis algorithm for the synthesis of reaction path networks is developed for industrial aromatic chemistry. Efficient screening is based on an implicit enumeration strategy exploiting logical reaction rules, constraints on structural attributes of

intermediates, and quantitative rate and selectivity targets. Reaction prediction techniques, including the use of quantitative structure-reactivity models, are proposed for systematic treatment of competing reactions.

2.3. Industrial Insights

Many specialty chemicals are part of a series of structurally similar compounds that evolves in response to new market opportunities, customer feedback, and maturing process technology. Process economics, scale of production, and availability of research resources often dictate that chemical products embody common classes of chemistry and process technology that can be optimized across multiple product lines. The goal of a reaction path network is to provide a set of alternative chemical reaction paths to improve the selection of process chemistry taking into account available raw materials, chemical conversions, intermediates, and multiple products. Industrial examples indicate that topological features present within the reaction path network can motivate development of certain lines of products and lead to exclusion of others. Potentially important topological features include

- precursors or chemical conversions used by multiple reaction paths;
- chemical reactions that produce usable by-products; and
- multiple reaction paths that converge on a single product.

This section provides industrial examples of such topological features in the context of aromatic chemistry. The examples qualitatively chart the process regimes which can give rise to the need for special reaction path structures. Such insights are useful in the preliminary screening of reaction path networks and motivate the choice of benzenoid compounds as a model system of chemistry.

Pharmaceutical, dye, and other fine chemical industries require hundreds of different raw materials and intermediates. Many subsets of these compounds are variations on a common theme, consisting of series of closely related products that differ only slightly in the functional groups and their relative orientations. Since the production of any one compound may be uneconomical due to low volumes (*i.e.*, high capital cost per unit volume; a good conceptual model is the six-tenths rule used in equipment cost correlations) and cyclical demand patterns, an integrated production strategy is often required. For example, development of a common precursor is almost always favorable to increase the produced volume of the precursor so as to lower its unit cost. Implementation of a common raw material (3-nitroisophthalic acid) is depicted in Figure 2.1 (Nobel Industries, 1993).

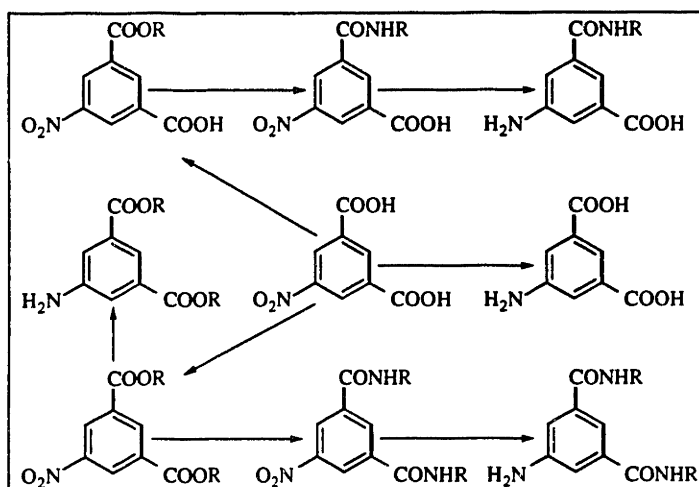


Figure 2.1. Integrated Synthesis Routes Starting with 5-Nitroisophthalic Acid (Adapted from Nobel Industries, 1993).

Also favorable for scheduling purposes and reducing capital costs is the integration of equipment requirements *vis-à-vis* chemical conversions transferable to several reaction paths. For example, the reaction path network depicted in Figure 2.1 requires only three functional group interconversions to produce the phthalic acids: esterification ($\text{COOH} \rightarrow \text{COOR}$); amino-de-alkoxylation ($\text{COOR} \rightarrow \text{CONHR}$); and reduction of the nitro group ($\text{NO}_2 \rightarrow \text{NH}_2$).

A second manner in which common precursors can be explored is through chemical conversions yielding several usable by-products. Such an approach has an advantage of potentially high useful conversion of the raw material, but involves higher separations costs. While the separation feasibility and cost is obviously dependent on the properties of the products and reagents, as a general rule the energy costs tend to become favorable at higher production volumes as compared to the previous example (Figure 2.1). A good example is the chlorination of phenol, depicted in Figure 2.2 (Muller and Caillard, 1985).

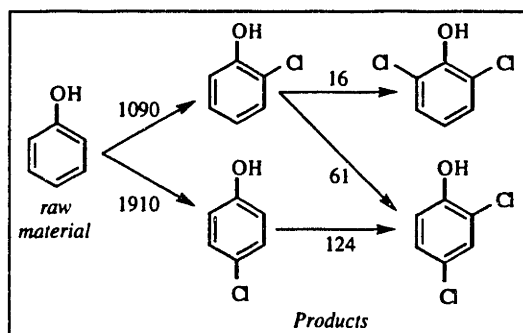


Figure 2.2. Generation of Multiple Products Through Chlorination. Numbers are rates of reaction relative to reaction of benzene. Adapted from Muller and Caillard (1985).

A number of process configurations can be developed to optimize the product distribution. If reaction in a single reactor is adequate, the product distribution can be controlled by the amount of chlorine added to the reacting system (MacMullin, 1953). If a single reactor does not provide sufficient selectivity, an improved distribution of products can be obtained through synthesis of reactor-separator networks.

In some cases it is desirable to produce and recover a by-product generated by an organic reagent rather than by the main reactant molecule. For example, when the reagent is very expensive, has a high waste disposal cost, or is difficult to recycle, it may be desirable to simply convert the reagent to a usable by-product. An illustrative candidate is the methyl-substituted aniline by-product produced by the Vilsmeier-Haack formylation (Figure 2.3).

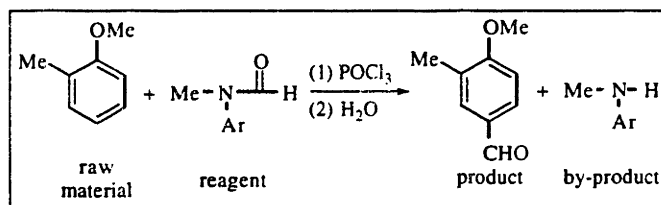


Figure 2.3. Exploiting Choice of Reagent as a Basis for Producing a Useful By-Product in the Vilsmeier-Haack Formylation.

This example shows that the formanilide precursor might be chosen so that the methyl-substituted aniline is a salable by-product. Though such a route might not be economically feasible for synthesis of the amine by itself (*e.g.*, the reagent may well be more expensive than the by-product), the route is still an interesting alternative since it avoids the additional reaction and separation costs of recovering the original formanilide reagent.

Rather than exploit a common precursor or multiple by-products, in certain processes opportunities arise to incorporate alternate precursors in a reaction path. A

number of examples are identified in the case study developed in chapter eight. Additionally, at least one proprietary agrochemical process has been developed (Steele, 1993) that can respond to fluctuations in commodity prices by switching among an array of raw materials and intermediates.

While the examples given above have been considerable achievements, no systematic approach has previously been developed to map out reaction path networks containing topological features such as those described above. In the following section, an applicable approach that assumes prior knowledge of target molecules and the set of available chemical transformations is proposed for industrial aromatic chemistry and could be extended to other classes of chemistry.

Motivating the study of industrial aromatic chemistry has been Hendrickson's (1971) examination of relevant reaction path design issues. A central part of the development is characterizing the multiplicity of synthesis routes arising from interconversion of structural forms. In addition to Ortho (O), Meta (M), and Para (P) forms for disubstituted compounds, Hendrickson (1971) defines six additional forms for higher substituted compounds (Figure 2.4).

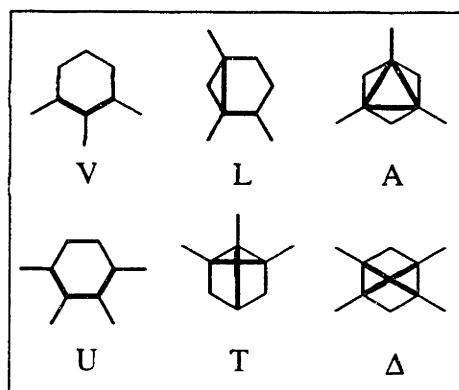


Figure 2.4. Higher Isomeric Structures Defined by Hendrickson (1971).

The interconversion of isomeric forms in reaction paths, achieved through functional group addition and removal reactions, must obey the generalization of Körner's Theorem (Hendrickson, 1971) shown graphically in Figure 2.5.

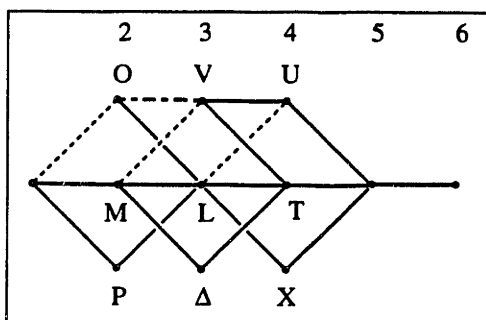


Figure 2.5. Interconnections Between Isomer Forms in Industrial Aromatic Chemistry. Adapted from Hendrickson (1971).

The equivalence of reaction sites due to the six-fold symmetry of the benzenoid substrate potentially allows reaction paths to be approached from several directions. Multiplicity of chemical reaction paths also tends to arise when a reaction produces several by-products, as is common in industrial aromatic chemistry. These facts, taken with the rich class of functional group interconversions, provide insight into why benzenoid compounds form a bridge between many basic and specialty chemicals.

2.4. Introduction to Retrosynthetic Analysis

Retrosynthetic analysis (Corey *et al.*, 1969) is a reaction path synthesis technique that starts with the target molecule and uses a knowledge of available chemical conversions (*transforms*, for short) to systematically generate candidate precursors. In effect, transforms are applied in reverse to define *retro-transforms*. A general discussion of retrosynthetic analysis is presented in chapter three. The mechanics of the technique consist of two steps: (i) evaluating the target molecule to find susceptible structures that might be prepared by an available transform; and (ii) applying an appropriate retro-transform to generate a precursor molecule (termed *retro-reaction*). As a simple example, consider the retrosynthetic analysis of 4-nitrotoluene (Figure 2.6).

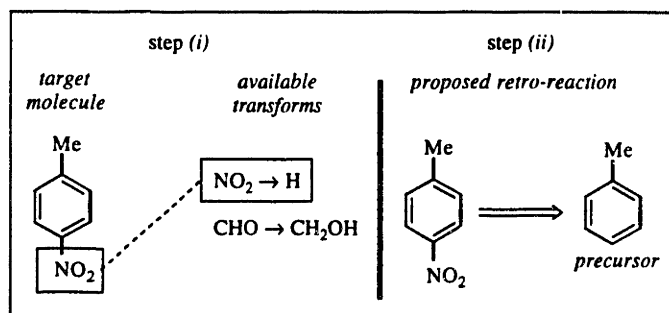
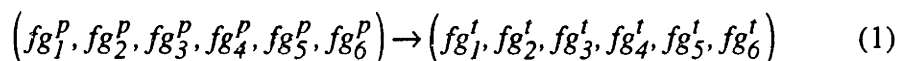
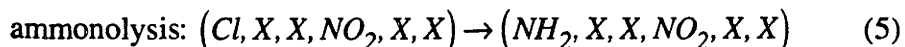
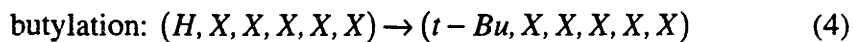
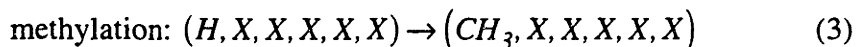
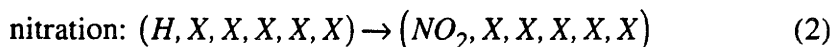


Figure 2.6. Simple Illustration of Matching Between Functional Groups on the Target Molecule and Available Transforms to Yield a Proposed Retro-Reaction. The symbol \Rightarrow is a commonly used designation of a retro-reaction.

Figure 2.6 is an example of the particular class of transforms with which this chapter is concerned, namely addition, substitution, and interconversion of functional substituents on benzenoid substrates. The transforms considered in this work generalize function group addition and functional group interchange on aromatic compounds. A substrate consists of up to six substituents attached at specified locations on a structural skeleton. Thus, the general transform has the form shown in Eq. 1.



where p denotes the precursor and t denotes the target molecule. Example transforms are given in Eqs. 2–5.



where X denotes an arbitrary value for the reaction site. A transform such as Eq. 1 is a template specifying the relevant functional groups and their relative orientations in a molecule (Figure 2.7).

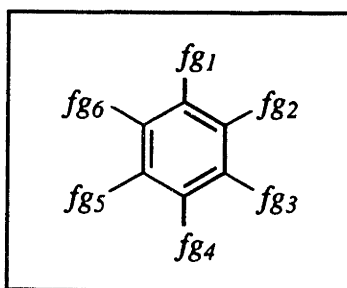
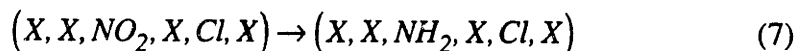
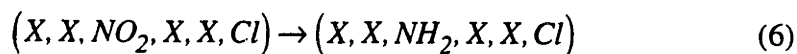


Figure 2.7. Mapping of Functional Groups Onto the Aromatic Nucleus.

It is important to recognize, however, that the mapping is not unique. For example, in the nitration of benzene substitution can occur with equal probability at any one of the six hydrogen atoms. At the same time, however, the ordering of functional groups must be consistent with the molecular structure. For example, in the ammonolysis of chloronitrobenzene (Eq. 5) a para relationship between the nitro group is necessary to make the site containing the chlorine atom susceptible to replacement by the amino group. Thus, Eq. 6 is equivalent to Eq. 5, but Eq. 7 is not.



Depending on the symmetry of the benzenoid substrate, there are up to twelve possible mappings consisting of six clockwise and six counter-clockwise rotations.

Using the transforms defined above, retrosynthetic analysis consists of applying the corresponding retro-transforms recursively to the target molecule and its precursors. As an example, the analysis of a synthetic musk is given in Figure 2.8. Considered in this example (without any attempt to screen the results) are the possible permutations of the nitration, methylation, and butylation retro-transforms (Eqs. 2–4).

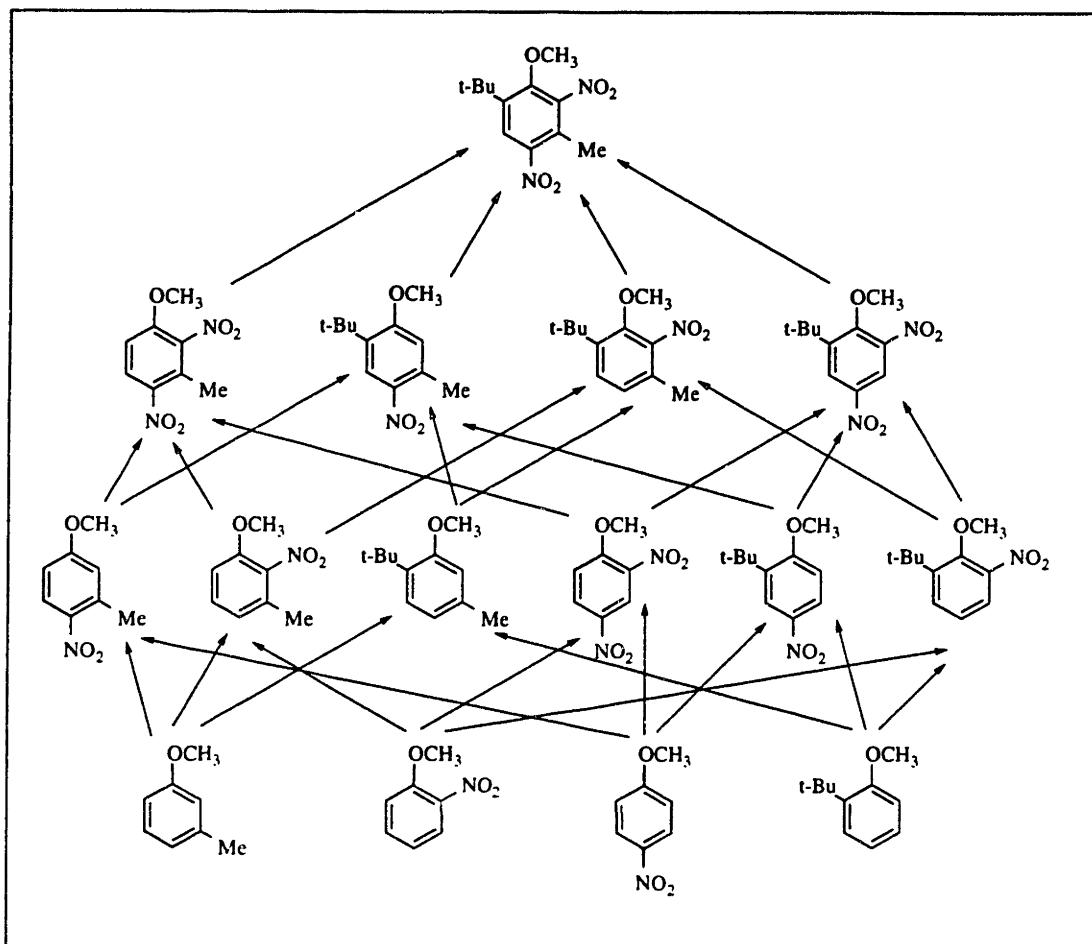


Figure 2.8. Illustration of a Portion of the Retrosynthetic Network Generated for Musk Ambrette.

The fourteen intermediates and twenty-four synthesis routes depicted in Figure 2.8 are created from various permutations of the three reactions. A much larger number of routes can be generated if the conversions from substituted phenol to anisole are considered, although the resulting phenolic compounds can exhibit undesirable handling properties such as strong acidity.

Systematic evaluation of reaction paths is a major outstanding problem in reaction path synthesis. Whereas retrosynthetic analysis generates sequences of "ideal" reactions, each consisting of a reactant molecule, a transformation, and a product molecule, real reactions can involve impurity reactions, by-products, reactions between solvent and the reagent, and a number of other side-effects. Additionally, it is possible that the rate of a candidate reaction is so low as to be negligible. In general, competing reactions as well as the intrinsic kinetics and thermodynamics of a reaction have a large impact on industrial scale-up. If only a few synthesis routes are being considered, it might be easy enough to

run the individual reactions in the laboratory. It is evident from the above example (Figure 2.8), however, that a combinatorial number of alternatives can be generated from permutations of a given number of alternate reactions. Also evident from the example is that a significant fraction of the routes are likely to be mere structural permutations with little industrial merit.

The goal of the retrosynthetic analysis algorithm developed in this chapter is to facilitate the rigorous identification of all promising routes satisfying a set of evaluation targets. Reaction path evaluation is addressed on two fronts. First, reaction prediction techniques are used in the evaluation of individual transforms. Second, results of the evaluations are compared with targets in an implicit enumeration strategy designed to "prune" infeasible portions of the reaction path network. Such an approach minimizes the combinatorial enumeration of reaction paths, thereby reducing computational requirements. It is noted that the representation of molecular structure and reactivity depicted here, while not the most general, is extendible to other classes, including heavier aromatic compounds. It is further believed that the algorithm developed below can be extended to a wide range of retrosynthetic transforms in addition to functional group addition and interchange.

2.5. Reaction Path Synthesis Testbed

A reaction path synthesis algorithm using retrosynthetic analysis has been created for certain classes of industrial aromatic chemistry. The algorithm partitions the synthesis of reaction path networks into two main components: (i) local retrosynthetic generation of sub-target molecules; and (ii) global implicit enumeration of alternate reaction paths. At the local level, retrosynthetic analysis is based on target and sub-target molecules containing up to six substituents on a skeleton of specified symmetry. At the global level, a general implicit enumeration strategy is developed. The strategy uses bounds derived from reaction evaluation, property estimation, or other sources to "prune" infeasible reaction subpaths. Bounds are implemented for the case of reaction prediction to motivate the development of reaction prediction strategies (section six).

2.5.1. Local Retrosynthetic Analysis

The local functionality is explained most directly by viewing a chemical reaction path as a graph consisting of nodes representing molecules and arcs representing retro-reactions. Starting with a "parent" node representing the target molecule, retro-transforms are applied to generate "child" nodes representing chemical precursors (Figure 2.9).

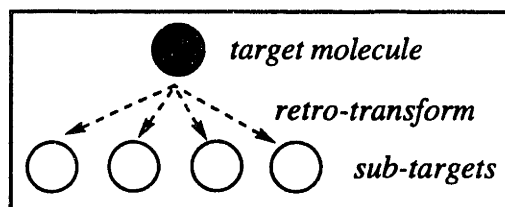


Figure 2.9. Graphical Representation of Target Molecule, Retrosynthetic Transformations, and Sub-Target Molecules. The dashed arrow is the designation of a retro-reaction used in this work.

In some cases multiple substituents in the target molecule are susceptible to a retro-transform, resulting in several precursors. For example, the retro-reduction of the two amine groups in chlorophenyldiamine (Figure 2.10) generates three precursors.

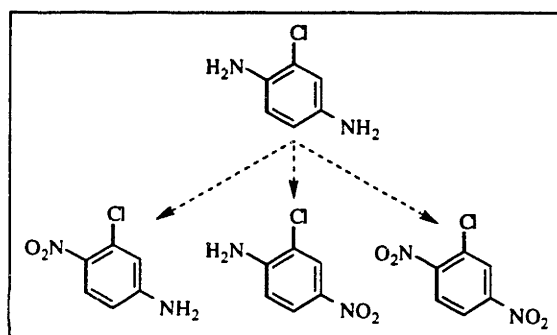


Figure 2.10. Generation of Precursors by Retro-Reduction of a Chlorophenyldiamine.

It is also important to recognize that the number of unique precursors is dependent on the symmetry of the target molecule. For example, the retro-nitration of trinitrotoluene at each susceptible site yields three precursors, but only two of these precursors are unique (Figure 2.11).

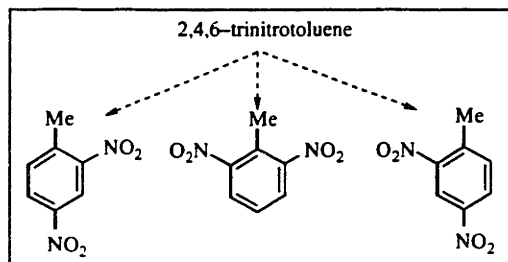


Figure 2.11. Presence of Equivalent Sites on Target Molecule Due to Symmetry in the Target Molecule.

In the situation depicted in Figure 2.11, the set of precursor molecules is compared and redundant nodes eliminated to form a unique matching between the set of retro-reactions and precursors.

2.5.2. Recursive Algorithm for Rigorous Synthesis of Chemical Reaction Paths

Reaction paths are generated through recursive application of local retrosynthetic analysis, as illustrated in Figure 2.12.

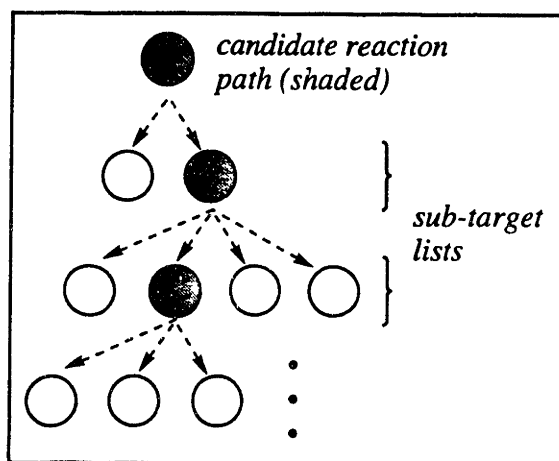


Figure 2.12. Recursive Application of Local Retrosynthetic Analysis to Identify Sub-Targets and Candidate Synthesis Routes.

In practice, the recursion process can generate a reaction path network containing hundreds or thousands of chemical compounds. The enumeration of all feasible pathways is a non-trivial problem requiring an algorithm to systematically track pathways that have already been identified and to look for new pathways. The key idea in the algorithm developed here is to characterize chemical reaction subpaths hierarchically in terms of recursively generated sub-target molecules. Each sub-target node has an attribute that takes one of two values: "under evaluation" or "completely fathomed." When a sub-target is completely fathomed, all subpaths leading to the node have been enumerated. The algorithm then "branches" to (*i.e.*, selects) the next sub-target molecule in a sub-target list and restarts the recursion. When all of the sub-targets in the list have been identified, the status of the sub-target at the next highest level (*i.e.*, closer to the target molecule) changes from "under evaluation" to "completely fathomed" and the algorithm branches to a new sub-target at the higher level.

The use of sub-targets as criteria for implicit enumeration of chemical reaction paths allows duplicate calculations to be avoided. Once a sub-target is completely

fathomed, there is no need to duplicate evaluations of subpaths and sub-targets below it (Figure 2.13).

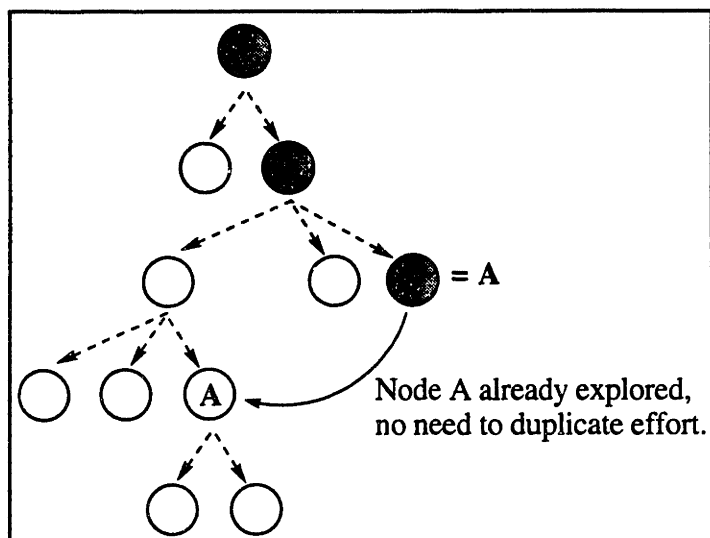


Figure 2.13. Referencing of a Previously Explored Node to Avoid Duplication of Effort.

The algorithm provides a natural stopping criterion: enumeration of all sub-target lists, which is equivalent to complete fathoming of all sub-targets in the global list. It is also useful to set a maximum chemical reaction path length. In addition to reducing the problem size, the length limit also reflects several practical considerations. For example, in specialty chemical manufacturing, synthesis routes beyond a specified length are generally unsuitable due to yield losses and economy of scale considerations. In particular, while desirable to have an inexpensive base chemical as a raw material, it is rarely desirable to produce a base chemical as a key intermediate within the process. Instead, production of the intermediate should be out-sourced.

It is noted that cycles connecting sub-targets may exist, but they are rarely of industrial significance. A strategy must account for them, however, because in cases such as reagent recovery they can be important topological features. Cycles in the network are handled in two ways. First, a maximum pathway length eliminates most, if not all, of the undesirable cycles. Second, the algorithm avoids infinite recursion by refusing to fathom a sub-target that is currently being fathomed.

2.5.3. General Strategy for Implicit Enumeration

Implicit enumeration of a reaction path network is the elimination of unsatisfactory subpaths without explicitly enumerating all of the individual transforms in those subpaths.

In particular, a reduction in the number of precursors at the local level leads to reduction in the global search space as shown in Figure 2.14.

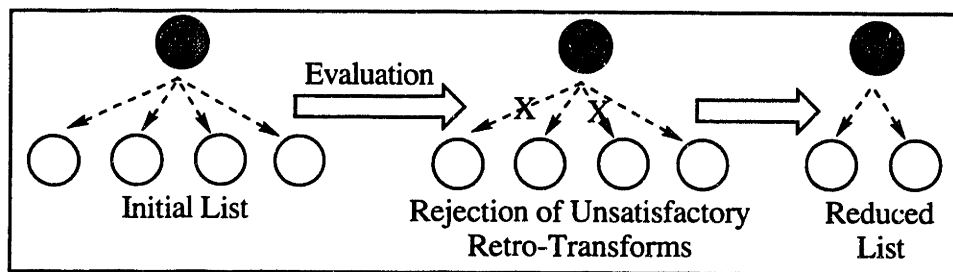


Figure 2.14. Use of Evaluation Criteria to Eliminate Infeasible Retrosynthetic Conversions From Consideration.

A good starting point for local analysis are criteria in the form of quantitative targets and logical reaction rules. Quantitative screening criteria consist of mathematical inequalities representing property or conversion targets to be satisfied. The general form of such inequalities is shown in Eq. 8.

$$A: a \geq A?, \quad (8)$$

where A is a vector of lower bounds that must be satisfied by one or more transforms and a are realized values. The purpose of the question mark (?) is to indicate that one or more of the lower bounds not being satisfied is taken as sufficient criterion for eliminating the corresponding transformations from consideration. Possible quantitative targets include lower bounds on the reaction rate and bounds on the chemical properties of sub-targets. Reaction rules incorporate screening criteria in the form of logical propositions, P , that must be satisfied. Their form is shown in Eq. 9.

$$P[\text{arguments}] = \text{True?}, \quad (9)$$

where *arguments* includes the type of transform as well as reactant and product substrate structures. As an example, a reaction rule might specify an interfering configuration of functional groups to be avoided. Examples are given below (general use of chemical logic in retrosynthetic analysis is described in chapter three).

A general strategy for implicit enumeration is developed to utilize criteria of the form of Eqs. 8 and 9 within the retrosynthetic analysis algorithm. To help explain the different levels at which the criteria are employed, Figure 2.15 introduces a formal notation for individual reaction paths and subpaths.

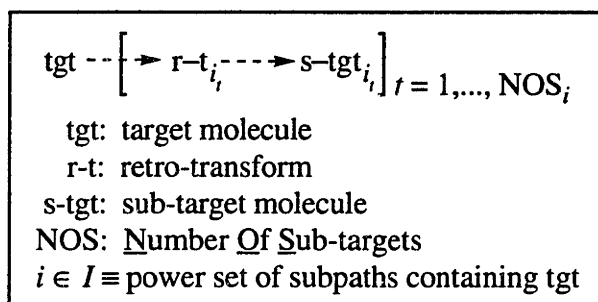


Figure 2.15. Retrosynthetic Depiction of a Reaction Path Consisting of a Target Molecule, Retro-Transforms, and Sub-Target Molecules.

The reaction paths of interest in the network are those that connect to the target (tgt) molecule. The retro-transforms and sub-targets present in such a reaction path are broken down into segments defined by Eq. 10.

$$\left(\text{r-t}_{i_t}, \text{s-tgt}_{i_{t+1}}, \text{r-t}_{i_{t+1}} \right) \quad (10)$$

The representation indicates several useful locations for imbedding screening criteria within a reaction path. These include individual retro-transforms, r-t_{i_t} , and sub-target molecules, s-tgt_{i_t} , as well as a complete transformation (Eq. 10). Quantitative measures can also be derived for reaction subpaths of the form depicted in Figure 2.15. Such bounds require passing evaluation information at a location i_t to subsequent levels $t^* \geq t$. It is important to recognize that elimination of a retro-transform or sub-target at a location i_t in a reaction path $i \in I$ effectively eliminates not one but all reaction subpaths at levels $t^* \geq t$.

Evaluation and screening of a reaction path consisting of three transforms are given in Figure 2.16. The sequence of evaluations to be performed is indicated by the letters (a-f). Quantitative bounds at the local level are given by the vector U_i (the index $i \in I$ is omitted since a single path is being considered; in other words, U_i is being written more compactly as U_t), which are compared with the predicted values u_i , $i = 1, 2, 3$. Similarly, the bounds applied to sub-paths are given by the vector V , compared with v_{11} , v_{12} , and v_{123} . Logical propositions are given by $P[\text{arguments}] = \text{True?}$, where *arguments* consist of the retro-transform and adjacent substrates.

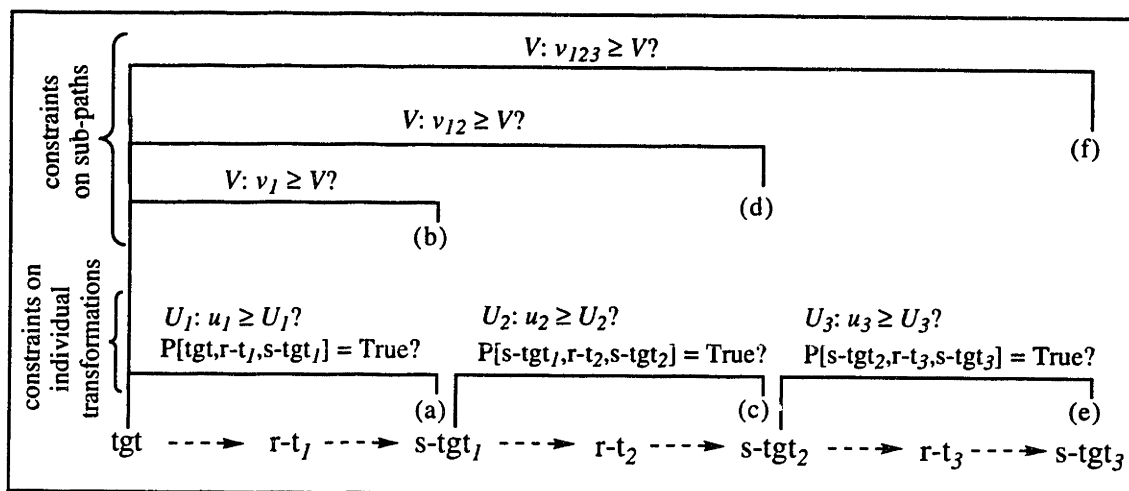


Figure 2.16. Hierarchy of Constraints in the Generation and Evaluation of a Chemical Reaction Path.

If any one of the retro-transforms or sub-targets can be excluded on the basis of screening criteria depicted in Figure 2.16, it should be clear that there is no need to continue fathoming the excluded sub-target. For example, if the reaction path in Figure 2.16 does not satisfy the targets at (d), there is no reason to look at r-t_3 or s-tgt_3 or to perform evaluations (e–f). Sub-target molecule s-tgt_2 is said to be "pruned." The fact that retrosynthetic analysis is directional—*i.e.*, it starts with the target molecule and works backwards to generate sub-target molecules—means that the first molecule or transformation that does not satisfy screening criteria provides a sufficient condition for branching to a new chemical reaction subpath. Figure 2.17 shows how the elimination of a transform from consideration can influence the generation of reaction subpaths.

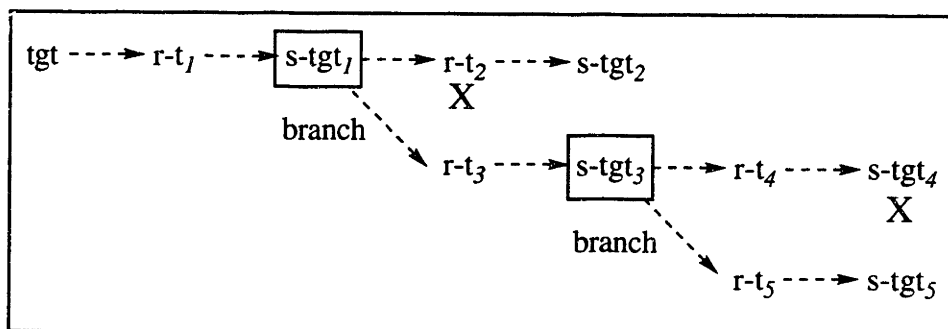


Figure 2.17. Illustration of How Pruning Can Eliminate Subpaths From Consideration. When sub-targets or retro-reactions are eliminated (X), it is possible to branch to a new synthesis route.

The most powerful set of targets is based on preliminary reaction prediction of the reaction rate and product fractions. It should be noted that the lower bounds do not have to correspond to the final selection criteria, but can be set much lower so that only manifestly unsatisfactory reactions are eliminated. Similarly, one can set a lower bound on the overall selectivity as well as the slowest acceptable step in a synthesis pathway consisting of several steps. In other words, a synthesis routes whose individual conversions pass the screening criteria can be eliminated if they involve too many low yielding steps. Such criteria are particularly useful as a way to control the length of a reaction path.

One of the reaction prediction-based targets is a uniformly applicable lower bound on the selectivity of the desired product. To illustrate the bound, assume a transform t_i creates a set of products, $\{j\} \equiv J_i$, with the desired product being denoted by $j^* \in \{j\}$. Given parameters y_j (e.g., relative reaction rates) that can be normalized to estimate product fractions, Eqs. 11 and 12 define a lower bound on the product fraction of the target molecule.

$$s_i \geq S^{LO}, \quad (11)$$

where

$$s_i \equiv \frac{y_{j^*}}{\sum_{j \in J_i} y_j}. \quad (12)$$

In establishing the lower bound, S^{LO} , it is important to recognize the potential role of major by-products for use in other pathways. Thus, rather than measure the selectivity of the reaction only for a single product, the bound represents a selectivity below which the separability costs outweigh possible benefits of by-product recovery. A second target similar to Eq. 11 is a transform-specific lower bound on the reaction rate. A reasonable lower bound can often be established for particular reaction types such as nitration and chlorination.

A third related target is the selectivity of the reaction pathway. A measure of the pathway selectivity can be obtained from intermediate product fractions starting with the target molecule and working backwards (Eq. 13).

$$\prod_{i \in \text{NOS}_j} s_i \geq \pi^{LO} \quad (13)$$

Equation 13 is representative of a class of targets in which computational results must be propagated to different levels in the reaction path network. A reaction subpath satisfying Eq. 13 in one reaction path will not be subsequently rejected in a second reaction path, since the synergy between the two reaction paths may ultimately justify the latter pathway.

Finally, a fourth class of screening criteria are reaction rules. As a simple example, diazotization of ortho amines is known to produce benzotriazoles (Hertel, 1985) instead of the desired diazo salts. Given two substituents present on the substrate, group_{g₁} and group_{g₂}, the logical form of the proposition is given in Eq. 14.

$$\begin{aligned} &\{(\text{transform} \neq \text{Sandmeyer}) \text{ OR} \\ &\left(\text{ORIENTATION}[\text{group}_{g_1}, \text{group}_{g_2}] \neq \text{ortho}\right) \text{ OR} \\ &\left(\text{GROUP_TYPE}[\text{group}_{g_1}] \neq \text{amine}\right) \text{ OR} \\ &\left.\left(\text{GROUP_TYPE}[\text{group}_{g_2}] \neq \text{amine}\right)\right\} = \text{True} \end{aligned} \quad (14)$$

The next section will present reaction prediction techniques for the synthesis of targets including Eqs. 11–14.

2.6. Reaction Prediction Techniques for Implicit Enumeration

Computing reasonable estimates for the bounds described above is a reaction prediction problem—*i.e.*, systematic prediction of the outcome of chemical reactions both in terms of the products that are generated as well as the rates and selectivities of the chemical reactions. Reaction prediction techniques applicable to industrial aromatic chemistry are explored in this section. The first relevant goal is assessing the set of products associated with a chemical reaction. In industrial aromatic chemistry, competitive substitution and rearrangements play an important role in the generation of by-products (Stock, 1968). As an illustrative example, consider the proposed nitration of toluene to yield a product *m*-nitrotoluene (Figure 2.18).

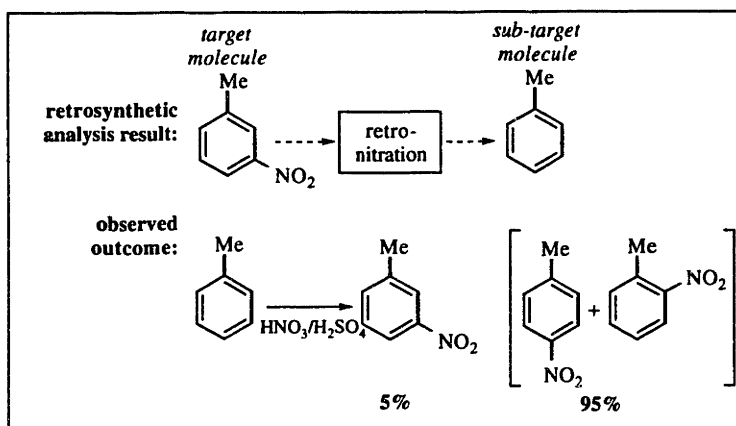


Figure 2.18. Retrosynthetic Nitration of *m*-Nitrotoluene that Generates Toluene as a Precursor. In practice, the presence of competing reactions reduces yield of the desired product below 5%.

There is nothing in the reactant's structure preventing the retrosynthetic analysis procedure from transforming the nitro group at the meta position of toluene. Any chemist knows, however, that the reaction strongly favors formation *o*- and *p*-nitrotoluene while yield of the meta substituted compound is generally below 5% (Stock, 1968).

A second major goal of reaction prediction should be the analysis of reaction rates and product fractions. For certain reaction classes quantitative structure-reactivity models are available that can estimate reaction rates or equilibrium constants relative to those of model compounds. A review of substrate structure-dependent models is given by Exner (1988). Models incorporating other dependencies are available, including solvent dependencies (Reichardt, 1990). Under the assumptions that competing reactions are kinetically limited and that the ratio of the rate laws yield a ratio of rate constants, the y_j parameters may be equated with rate constants. Further assuming that the substituent effects are logarithmically additive (Taylor (1990) discusses of the scope of this assumption for different reactions), an *additivity principle* that uses partial rate factors for mono and disubstituted benzenes can be derived from Hammett's linear free-energy relationship (Stock, 1968). Where valid, the principle allows reactivities at different sites to be assessed. A partial rate factor, denoted by k/k_{ref} , is the dimensionless ratio of a reaction rate to that of an appropriate reference reaction. The additivity principle is given by Eq. 15 for a single site on a benzenoid compound with two ortho substituents, two meta substituents, and a para substituent. The reference rate constant often corresponds to the reaction of benzene or toluene.

$$y = \frac{k^{site}}{k_{ref}} = \left(\frac{k_o^1}{k_{ref}} \frac{k_o^2}{k_{ref}} \right) \left(\frac{k_m^1}{k_{ref}} \frac{k_m^2}{k_{ref}} \right) \left(\frac{k_p^1}{k_{ref}} \right) \quad (15)$$

Often insufficient data or model limitations hinder the prediction of reaction rates for competing reactions. Such reactions can proceed by different mechanisms and lead to structurally different products. An illustrative example can be found in the Sandmeyer transform that converts an aniline to the corresponding benzonitrile (Figure 2.19; this transform is one of several possible Sandmeyer transforms and includes the diazotization step to simplify the presentation); to avoid confusion, it should be noted that diazotization fails to obey a Hammett-type relationship (Exner, 1972).

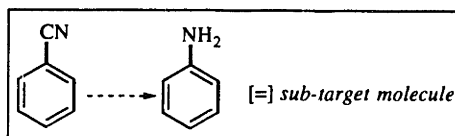


Figure 2.19. Application of a Sandmeyer Retro-Transform to Obtain Aniline from Benzonitrile.

When the aniline precursor reacts, several by-products are observed due to competing chemical transformations, particularly isomerization of the cis- adduct (Hagedorn, 1985; Figure 2.20).

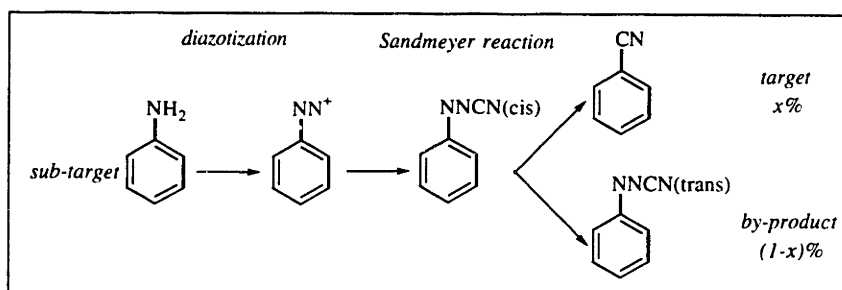


Figure 2.20. Product Distribution Associated with the Diazotization-Sandmeyer Reaction Sequence.

Although the mechanisms by which the competing transformations occur are different (Hagedorn, 1985), they operate on a common reactant molecule. Thus, approximate y_j parameters can be extracted from experimental data through a proportionality factor with partial rate factors for the desired cyanation.

The diazotization-Sandmeyer reaction sequence illustrates reaction mechanism generation and analysis in the solution of a reaction prediction problem. In general, analysis tasks tend to be specific to the reaction of interest, requiring special subroutine codes containing relevant data, models, and reaction rules. Therefore, a two part strategy is adopted in the computational implementation (details of the computational implementation are presented below): (i) the generation of the product set from available transforms; and (ii) analysis of the chemical reaction using specialized analysis modules. The next section develops a general relationship between a chemical reaction and a

template of chemical transforms to facilitate the range of analysis under consideration. Following the development, example analysis modules are presented for three cases.

2.6.1. Formal Representation of Competing Sites and Competing Reactions

The next step is to define chemical reactions in terms of basic chemical transforms to allow preliminary assessments of competing reactions and by-product formation. As discussed above, it is important to keep in mind that the types of products that can be formed depend both on the reaction and on the reacting substrate itself. As a result, it is useful to define a chemical reaction as not one but a set of chemical transformations, each of which can potentially correspond (or not) to the reverse of a retrosynthetic transformation applicable to one or more sites on the substrate molecule. Figure 2.21 illustrates the definition. The asterisks (*) in Figure 2.21 indicate that the computational implementation allows an arbitrary subset of the specific transforms to be used as retro-transforms.

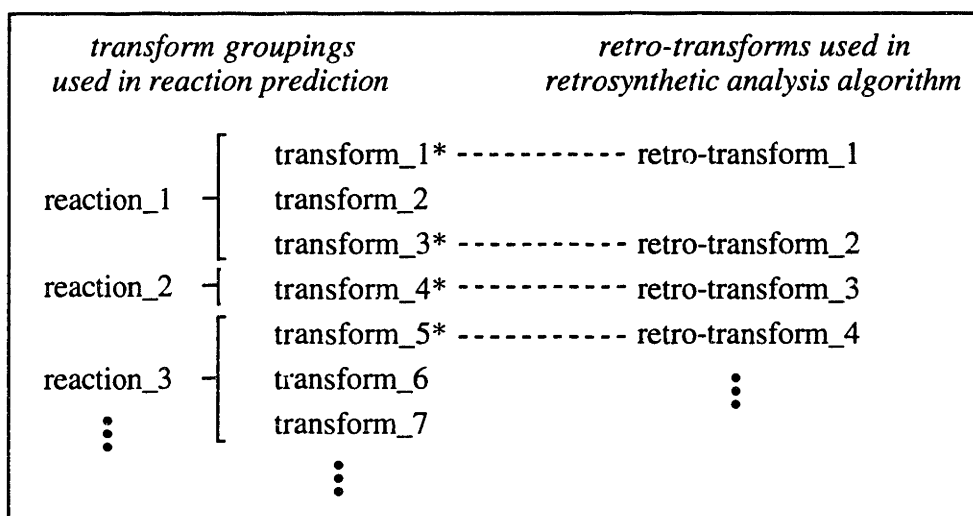


Figure 2.21. Illustration of How the Set of Transformations Associated with a Chemical Reaction Can Produce Multiple Transformations of Functional Groups and Therefore Yield Multiple Products.

So as to provide an unambiguous definition of retro-transforms at the algorithmic level, the individual transforms must be applied in parallel (serial application would involve additional protocol) to the reactant substrate. Important features of a mechanism such as intermediate steps and multiple substitutions are handled through explicit enumeration of the reaction subpaths as "aggregate" chemical transforms. A further consideration of the diazotization-Sandmeyer reaction sequence serves to clarify the idea of aggregate transforms and to demonstrate the generality of the representation.

The main elements of the mechanism are restated on the left side of Figure 2.22. Based on the mechanistic elements, two aggregate chemical transforms are defined to convert the amino group into the two main products.

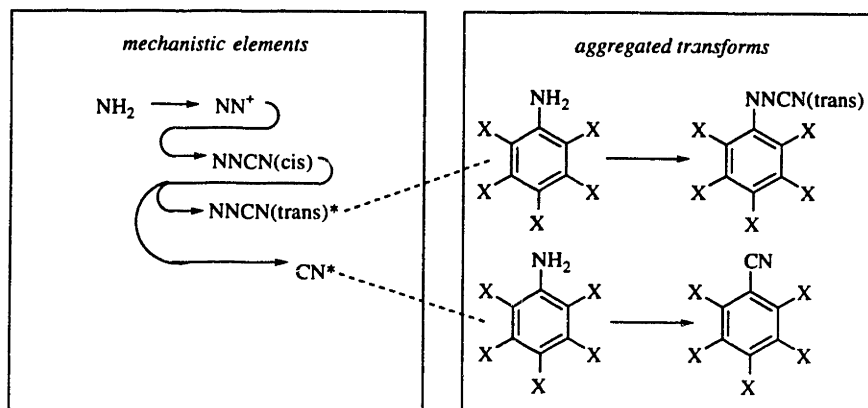


Figure 2.22. Main Steps in the Diazotization-Sandmeyer Reaction Sequence and the Corresponding Retro-Transforms.

A second situation that might be encountered is a substrate with two (or more) reacting groups. For example, in the reduction of a nitro group to an amine, a single reduction transformation will yield the nitroaniline products, but will miss phenyldiamine products. If the formation of such compounds is possible, additional aggregate transforms are required to generate the missing subpaths. For meta oriented dinitro compounds, the relevant aggregate transforms are given in Figure 2.23.

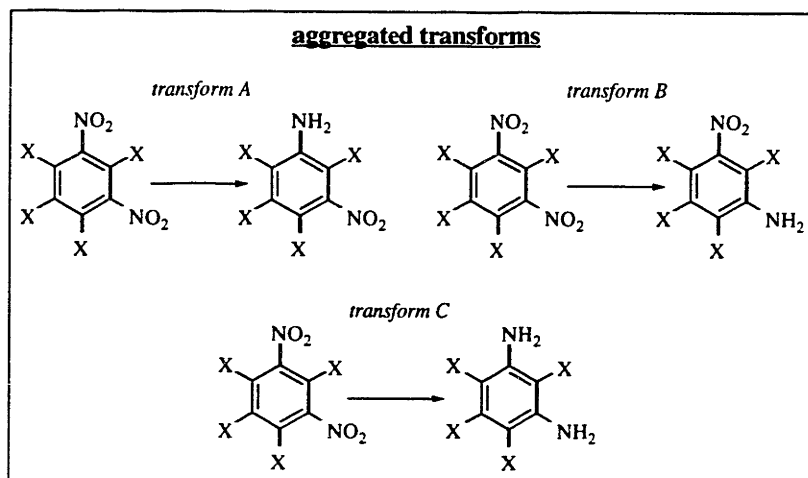


Figure 2.23. Additional Aggregated Transforms Applicable to the Nitro Group Reduction to Account for Dinitro Reactants.

Several implications of the reaction representation are worth mentioning. First, the presence of multiple transforms in a chemical reaction represents opportunities for integrating the process chemistry. Furthermore, as in the case of retrosynthetic analysis, the same transform can produce several products. Second, the representation makes it possible to take an abstract object such as a retro-transform and associate with it experimental data for specific reagents and reaction conditions. This connection is useful because data may be available only for discrete values. Thus, instead of always being forced to assume a fully general model, it is possible to define a new reaction at each discrete choice of reaction conditions.

2.6.2. Synthesis of Transform Networks

2.6.2.1. Monosubstitution Transform Networks: Homogeneous Nitration

Monosubstitution reactions are those for which polysubstitution at available reaction sites tends to occur on a much slower time-scale than the initial substitution reaction. A good example of a monosubstitution reaction is the nitration of an aromatic substrate by electrophilic aromatic substitution. Net stoichiometry is given in Eq. 16 for the case of homogeneous nitration in acetic anhydride.



Experimental partial rate factors for homogeneous nitration (2 M HNO₃/Ac₂O; 25°C) of nitrobenzene are 2.88E-6 (ortho), 4.17E-5 (meta), and 2.96E-7 (para) (Taylor, 1990). The rate factors show that addition of a nitro group to the aromatic ring tends to deactivate the substrate to further substitution. Using the partial rate factors given in Table 2.1 (the calculation of the parameters is discussed in chapter eight), predicted relative rates for the reaction of acetanilide are given in Figure 2.24.

Table 2.1. Partial Rate Factors for the Estimation of Chemical Reaction Rates. Chlorination corresponds to molecular chlorination in acetic acid; nitration corresponds to 2 M HNO₃/Ac₂O; 25°C.

Chlorination			
Substituent	<i>ortho</i>	<i>meta</i>	<i>para</i>
Cl	0.019	6.54E-4	0.0654
CN	5.38E-7	1.71E-6	1.35E-7
NO ₂	6.28E-9	3.03E-8	1.12E-9
NHAc	6.1E5	0.15	2.5E6

Nitration			
Substituent	<i>ortho</i>	<i>meta</i>	<i>para</i>
Cl	0.13	8.4E-4	0.13
CN	9.28E-5	4.36E-4	2.15E-5
NO ₂	2.88E-6	4.17E-5	2.96E-7
NHAc	8921	0.33	5248

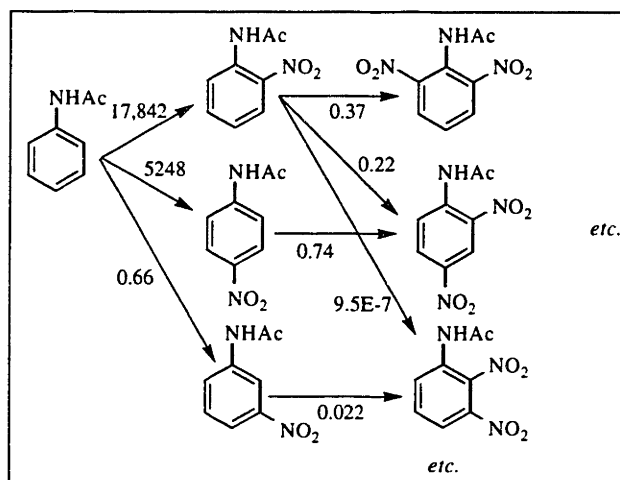


Figure 2.24. Acetanilide Nitration Rates Predicted by the Additivity Principle.

The magnitudes of relative rate constants in Figure 2.24 suggest that a reasonable estimate of the product distribution can be obtained by considering only monosubstitution among available reaction sites. Generally, multiply substituted compounds will be produced in 1–5% yield, levels that are potentially significant from a chemical engineering standpoint but which are almost impossible to quantitatively predict. It is further expected that additional by-products will be formed in the 1–2% range that are not accounted for in any of the reaction modeling.

It is concluded that analysis of competing reactions on the initial substrate makes the best use of limited available information. The main components of a simple reaction prediction analysis subroutine for homogeneous nitrations is given in Figure 2.25.

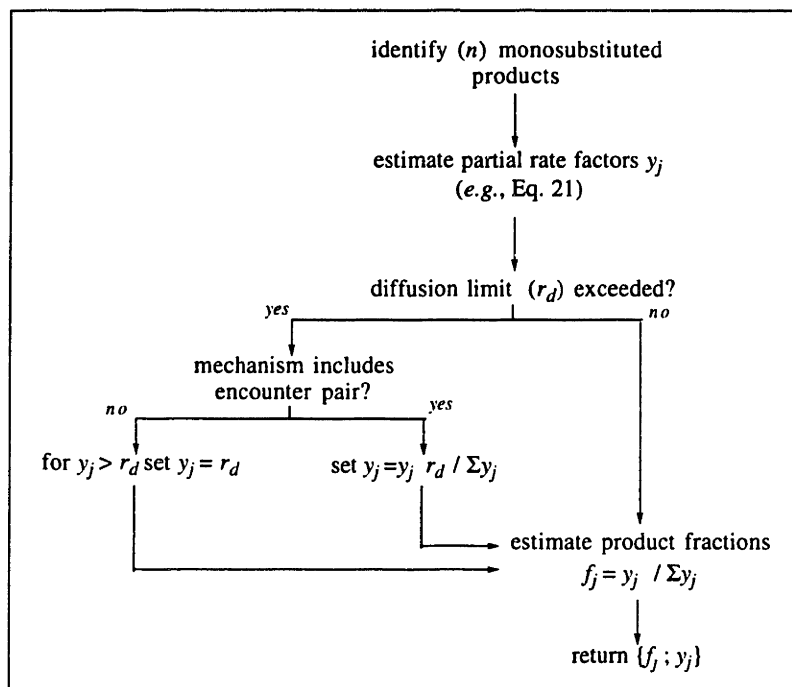


Figure 2.25. Main Steps in the Synthesis and Evaluation of Monosubstitution Transform Networks.

Di- and higher substituted substrates are discarded, leaving n monosubstituted substrates for analysis within Figure 2.25. Partial rate factors are estimated using available data and assuming the additivity principle (although the principle is subject to considerable deviations for nitration as discussed by Taylor (1990)). Often activated substrates undergo rates of nitration in excess of the diffusion limitations, particularly in mixed acid nitrations (Cox, 1972a-b). Diffusion limits can be checked and necessary corrections made to predicted rate factors. It should be noted that over some ranges of reagent concentrations the diffusion limitation might not impair the selectivity relationship, such as when the mechanism involves an encounter pair (Cox, 1972a,b). Finally, estimated partial rate factors and product fractions are returned to the retrosynthetic analysis algorithm for use in evaluation feasibility targets described above. The analysis in Figure 2.25 applied to the nitration of acetanilide is given in Figure 2.26.

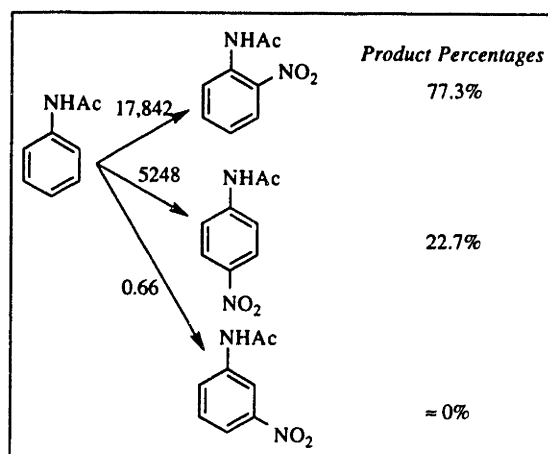


Figure 2.26. Reduction of the Complex Reaction Mechanism to a Selectivity Analysis Based on Predominance of Ortho and Para Monosubstitution.

Significant assumptions underlie the proposed analysis. First, the use of partial rate factors assumes that the reaction proceeds by the same rate-limiting step regardless of which combinations of substituents are present on the substrate. The assumption can lead to significantly incorrect predictions of selectivity when a change in mechanism occurs. For the analysis to account for such situations, reaction rules would be needed to screen deleterious substituent effects on the rate limiting step and systematic corrections made to the data. Second, the additivity treatment or related model is assumed to be valid for the proposed reaction. While nitration approximately obeys the Hammett-type relation for single substituents, the additivity treatment can break down when multiple activating substituents are present (Taylor, 1990).

Additional competing reactions can also be incorporated in a monosubstitution transform network. In the case of homogeneous nitration in acetic anhydride, direct acetylation of the aromatic ring can sometimes occur due to a reaction involving the Wheland intermediate. Additionally, solvent-reagent degradation reactions often need to be treated at the specific reagent concentrations and independently of the input chemical transforms. For example, acetic anhydride is susceptible to degradation pathways either due to impurities or through its reaction with nitric acid (Paul, 1958).

Finally, it is important to recognize that if degradation reactions do not play a role in the reaction mechanism, the product fraction calculations obtained are rigorous upper bounds on the true values to within model accuracy.

2.6.2.2. Polysubstitution Transform Networks: Sandmeyer Reaction

A second class of reaction mechanisms involves both parallel and multi-step conversions that can be assumed to proceed rapidly and completely. Under such circumstances the

multi-step feature of the problem can be handled by propagating product fractions. A simple example of such a network has been illustrated for the case of the Sandmeyer reaction sequence consisting of the diazotization followed by cyanation of an aniline substrate (Figure 2.20).

2.6.2.3. General Polysubstitution Transform Networks: Chlorination

As a third example, a more general polysubstitution transform network is given. Electrophilic aromatic chlorination substitutes a hydrogen atom with a chlorine atom on the aromatic nucleus. Unlike the nitration reaction, in which the addition of a nitro group tends to deactivate the aromatic substrate to allow easy control of the level of substitution, the partial rate factors in Table 2.1 indicate that multiple substitution is often a significant factor the chlorination reaction. As a result, more careful control is required for the chlorination reaction, especially when dealing with activated substrates. At the same time, the ability to achieve multiple substitutions in a relatively short amount of time (hours) makes chlorination a more suitable reaction for exploiting by-products as feeds to other reaction paths. In either case, for chlorination both parallel and multi-step paths must be considered to obtain an accurate estimate of the product distribution.

An analysis module should construct the transform network from the supplied transforms and product set. Assuming mass action kinetics (MacMullin, 1948), rate constants for the network can be estimated either from the additivity principle or experimental data. Closed-form solutions to the appropriate mass action equations are used to find the amount of chlorine that must be added to maximize the concentration of one of the products. The resulting solution is suitable for use in preliminary mass balances and in the analysis of by-product distributions when chlorinations are encountered during retrosynthetic analysis. The product fractions, the chlorine required, and the partial rate factors can be returned from the analysis.

The chlorination reaction of phenol provides a good example. This reaction produces a distribution of mono- and disubstituted phenols. The reaction network is depicted in Figure 2.27.

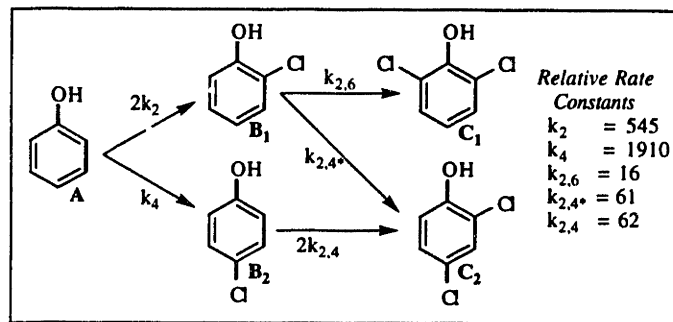


Figure 2.27. Reaction Network for Batch Chlorination of Phenol. Relative rate constants are for Iron-catalyzed chlorination with respect to rate for benzene.

The differential-algebraic equations describing the mass action kinetics for this system are given by Eqs. 17–27, where concentrations are expressed as mole fractions. It should be noted that the dependency on chlorine concentration is not shown in the rate expressions because it will cancel out in manipulations to follow.

$$-\frac{dA}{dt} = (2k_2 + k_4)A \quad (17)$$

$$-\frac{dB_1}{dt} = -2k_2A + (k_{2,6} + k_{2,4^*})B_1 \quad (18)$$

$$-\frac{dB_2}{dt} = -k_4A + 2k_{2,4}B_2 \quad (19)$$

$$-\frac{dC_1}{dt} = -k_{2,6}B_1 \quad (20)$$

$$-\frac{dC_2}{dt} = -k_{2,4^*}B_1 - 2k_{2,4}B_2 \quad (21)$$

$$A + B_1 + B_2 + C_1 + C_2 = 1 \quad (22)$$

$$B_1 + B_2 + 2(C_1 + C_2) = X \quad (23)$$

$$B_1, B_2, C_1, C_2 = 0 \text{ for } t \leq 0 \quad (24)$$

$$A = 1 \text{ for } t \leq 0 \quad (25)$$

$$0 \leq A, B_1, B_2, C_1, C_2 \leq 1 \quad (26)$$

$$0 \leq X \quad (27)$$

The variable X is the amount of chlorine consumed by the reaction at any point in time. Equation 22 is an overall material balance. Equation 23 gives the amount of chlorine consumed as a function of overall generation of products. Equations 24 and 25 give the initial conditions for the system. Equations 26 and 27 are bounds on mole fractions and chlorine consumed.

The system as written consists of twelve variables and eleven linearly independent equations (Eq. 22 can be obtained by adding together the differential equations and solving with Eq. 24 and 25 as initial conditions). A solution to Eqs. 17–27 will give eleven of the variables as a parametric function of the twelfth. For example, the equations could be solved numerically to obtain the concentration profiles as a function of time, t . The time dependency, however, is not the most useful parametric variable. In practice, the batch reaction time is controlled by the rate of introduction of chlorine. Furthermore, MacMullin (1948) has demonstrated that the product composition depends only on the amount of chlorine consumed independently of the rate of addition. Therefore, a more useful approach to the problem is to study the system in terms of the chlorine consumption, X .

The first step in the solution procedure is to eliminate the time variable from the problem by dividing Eqs. 18–21 by Eq. 17, resulting in Eqs. 28–31.

$$\frac{dB_1}{dA} = \frac{-2k_2A + (k_{2,6} + k_{2,4*})B_1}{(2k_2 + k_4)A} \quad (28)$$

$$\frac{dB_2}{dA} = \frac{-k_4A + 2k_{2,4}B_2}{(2k_2 + k_4)A} \quad (29)$$

$$\frac{dC_1}{dA} = \frac{-k_{2,6}B_1}{(2k_2 + k_4)A} \quad (30)$$

$$\frac{dC_2}{dA} = \frac{-k_{2,4*}A - 2k_{2,4}B_2}{(2k_2 + k_4)A} \quad (31)$$

Equations 30–33 are recognized as first order homogeneous differential equations, which are solvable in closed form by making substitutions of the form $W \cdot A$ for each of the composition variables B_1 , B_2 , C_1 , and C_2 . The solutions are given in Eqs. 32–38.

$$B_1 = a(A^b - A) \quad (32)$$

$$B_2 = \bar{a} \left(A^{\bar{b}} - A \right) \quad (33)$$

$$C_1 = \frac{-a k_{2,6}}{2k_2 + k_4} \left[\frac{A^b - 1}{b} - (A - 1) \right], \text{ where} \quad (34)$$

$$a = \frac{2k_2}{(2k_2 + k_4) - (k_{2,6} + k_{2,4*})} \quad (35)$$

$$b = \frac{k_{2,6} + k_{2,4*}}{2k_2 + k_4} \quad (36)$$

$$\bar{a} = \frac{k_4}{(2k_2 + k_4) - 2k_{2,4}} \quad (37)$$

$$\bar{b} = \frac{2k_{2,4}}{2k_2 + k_4} \quad (38)$$

Relationships for C_2 and X follow from solution of Eqs. 22 and 23.

Equations 32–38 give the composition profiles as a function of A explicitly and X implicitly. In the operation of a real process, it is usually desired to achieve an optimal product distribution that might consist of maximizing the concentration of a single product, minimizing the generation of an undesired by-product subject to additional production target constraints, or optimizing some affine combination of product concentrations. Since A decreases monotonically as a function of X , the one degree of freedom in the reacting system can be viewed as A , which can be substituted to determine X . Maximizing B_j with respect to A yields Eq. 39.

$$\frac{dB_j}{dA} = 0 \Rightarrow \frac{d(A^b - A)}{dA} = bA^{b-1} - 1 = 0 \quad (39)$$

Rearranging Eq. 41, the value of A maximizing B_j is obtained (Eq. 40).

$$A = b^{1/(1-b)} \quad (40)$$

Similar expressions can be obtained for B_j or other species of interest. Whichever species is the sub-target under consideration, the maximum attainable fraction is a reasonable basis for application of product fraction targets. In the implementation of the

reaction, the distribution of product fractions could be optimized to meet the needs of the overall reaction path network.

2.7. Computer Implementation

The retrosynthetic analysis algorithm has been implemented as a hybrid of FORTRAN 77 and C routines. The code has been implemented on the DEC Alpha/AXP and the Cray C90. The use of FORTRAN has facilitated vectorization of certain routines. Organization of the computer code is depicted in Figure 2.28. Major subroutines are denoted by "routine name." A listing of the computer code is given in the appendix six.

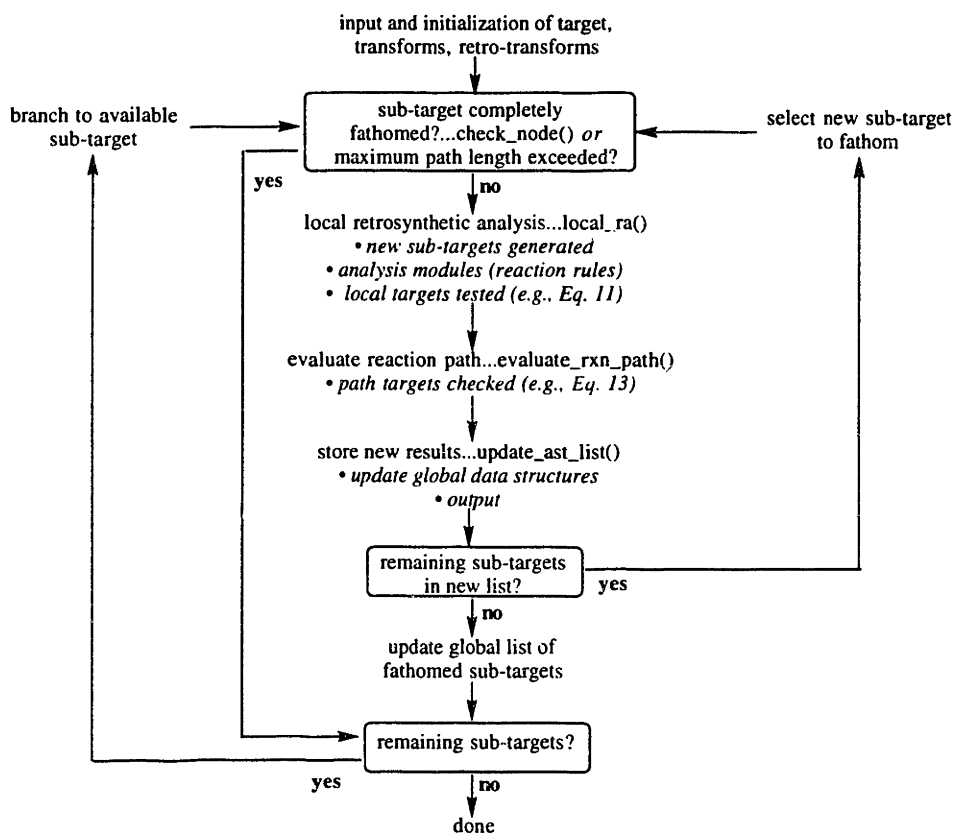


Figure 2.28. *Ra_Driver*: Overview of Retrosynthetic Analysis Algorithm Implementation.

Strategies for avoiding duplicate calculations (section 5.2) and implicit enumeration (section 5.3) are implemented through the feedback loops (Figure 2.28). The status of a sub-target molecule is assessed initially in order to determine whether there is a need to perform local retrosynthetic analysis. If the sub-target is either "completely fathomed" or in a reaction path of maximum length (section 3.5.2), then local analysis is by-passed and a new sub-target identified. When local analysis is performed

(section 5.1), a new list of sub-targets is generated. As long as the "current" list remains non-empty, new sub-targets continue to be generated. When an empty list is encountered, the parent sub-target assumes the "completely fathomed" status and the branching strategy (section 3.5.3) invoked to identify a new sub-target. The algorithm terminates when the set of available (*i.e.*, "under evaluation") sub-targets is empty.

The computer implementation of local retrosynthetic analysis and reaction prediction is depicted in Figure 2.29.

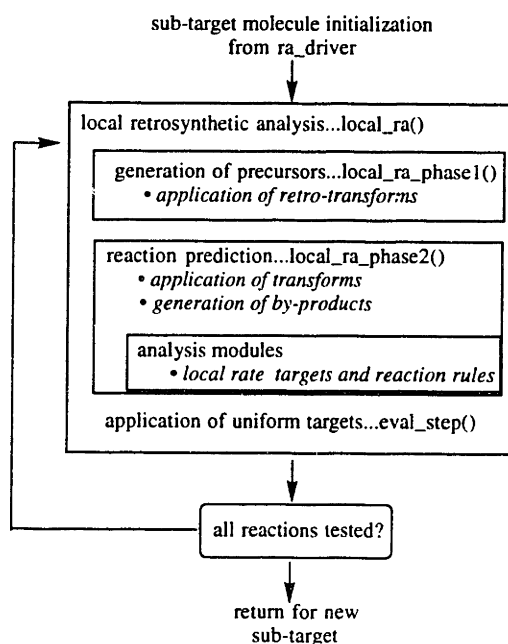


Figure 2.29. Local_Ra: Overview of Local Retrosynthetic Analysis and Reaction Prediction Implementation.

Local and pathway targets (Figure 2.16) are distributed hierarchically according to specificity of the reaction type. Low-level targets, consisting of minimum rate bounds (Eq. 9) and reaction rules (Eq. 14) (section 5.3), are tested in reaction prediction analysis modules within local retrosynthetic analysis. The next level of targets are product fraction targets defined by Eq. 14. These are performed within the local retrosynthetic analysis module independently of specific analysis modules (section 6). Failure to satisfy any of these targets results in the candidate sub-target being eliminated from the sub-target list. Remaining sub-targets are returned from local retrosynthetic analysis and tested by the reaction subpath evaluation module using the subpath product fraction bound (Eq. 13).

The flow of information among different components of the algorithm requires generating and maintaining databases of unique sub-target and by-product molecules throughout the analysis. The data structure that characterizes the results is a list of unique unit subpaths each of the form in Eq. 10 augmented by the set of by-product molecules.

The molecular objects "precursor" and "product" point into a global sub-target list while "by-products" each point to a global by-product list. Each object also points at a data structure containing quantitative information. Rate and product distribution information is associated with each reaction. Fractional values for conversion are attached to each molecular object.

2.8. Example: Synthetic Musk Intermediate

The retrosynthetic analysis algorithm is illustrated for the musk ambrette precursor 2-tert-butyl-4-nitro-5-methylanisole (an enumeration of reaction paths to musk ambrette is given in Figure 2.8). The chemical conversions nitration (**N**), Friedel-Crafts butylation (**Bu**), and methylation (**M**) are considered. The product fraction bound is 50% and relative rate bound 0.001 for each reaction. A reaction rule is defined restricting reaction path generation to disubstituted and higher substituted compounds. Disubstituted raw materials should be the most economical given the larger number of process steps required if a monosubstituted raw material is employed.

For simplicity the simple logarithmic additivity relationship (Eq. 15) is used for each of the reactions under consideration. Partial rate factors estimated from appropriate Hammett-type parameters and observed ortho:para ratios are given for the substituents in each orientation in Tables 2.2–2.4.

Table 2.2. Estimated Partial Rate Factors for Homogeneous Nitration (2 M HNO_3 / Ac_2O , 25°C). From Stock (1968), Hogget et al. (1971), and de la Mare and Ridd, (1959).

Substituent	<i>ortho</i>	<i>meta</i>	<i>para</i>
NO_2	2.00E-4	9.55E-5	1.82E-5
OCH_3	1.87E+4	5.01E-1	4.79E+4
$\text{CH}(\text{CH}_3)_3$	2.40	3.98	3.63E+1
CH_3	3.25E+1	2.63	4.79E+1

Table 2.3. Estimated Partial Rate Factors for Friedel-Crafts Methylation (2 M CH_3Br / GaBr_3 , 25°C). From Norman (1965).

Substituent	<i>ortho</i>	<i>meta</i>	<i>para</i>
NO_2	1.66E-03	4.38E-03	1.66E-03
OCH_3	2.78E+02	6.67E-01	5.57E+02
$\text{CH}(\text{CH}_3)_3$	0.00E+00	2.25E+00	8.23E+00
CH_3	7.84E+00	1.76E+00	9.67E+00

Table 2.4. Estimated Partial Rate Factors for Friedel-Crafts Butylation (2 M *t*-BuBr / GaBr₃, 25°C). From Attina et al. (1977).

Substituent	<i>ortho</i>	<i>meta</i>	<i>para</i>
NO ₂	6.03E-03	1.31E-02	6.03E-03
OCH ₃	1.56E+02	7.24E-01	1.56E+02
CH(CH ₃) ₃	0.00E+00	1.91E+00	5.38E+00
CH ₃	0.00E+00	1.76E+00	6.12E+00

Initial generation of sub-targets yields three candidates, (a-c) (Figure 2.30). The proposed methylation, (c), fails to satisfy the product fraction bound. Comparison of competing sites in nitration of 2-*tert*-butyl-5-methylanisole gives a target product fraction of approximately 72%.

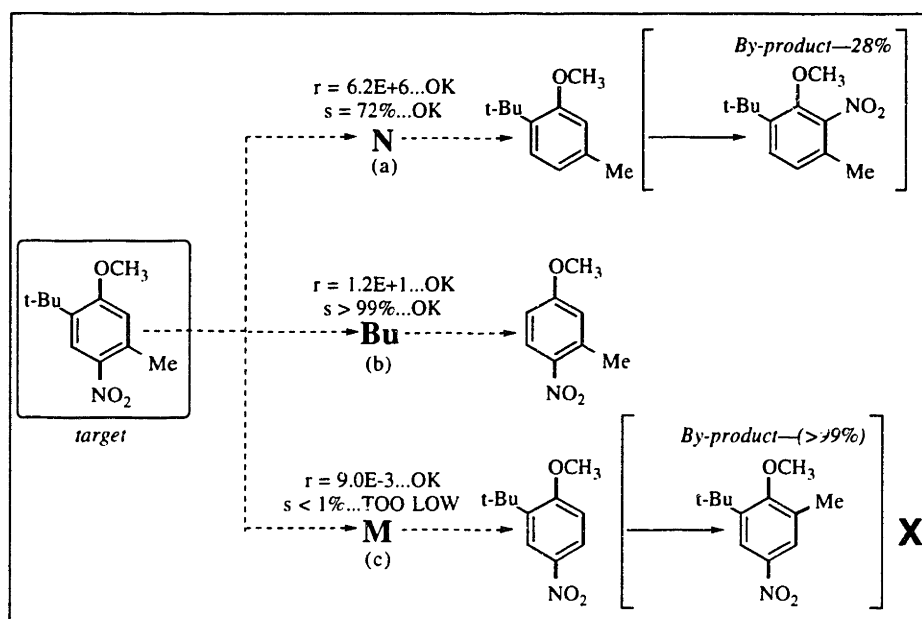


Figure 2.30. Initial Fathoming of Target Molecule (a-c). The symbol "r" is the partial rate factor; "s" is the production fraction.

The sub-target 3-methyl-4-nitroanisole is fathomed next (d-e) (Figure 2.31). The methylation route is again eliminated from consideration with the product fraction bound. The remaining route terminates in disubstituted 3-methylanisole. A nitration route produces three main products, with the desired sub-target being produced with a selectivity of about 50%. Thus, the nitration is retained for consideration.

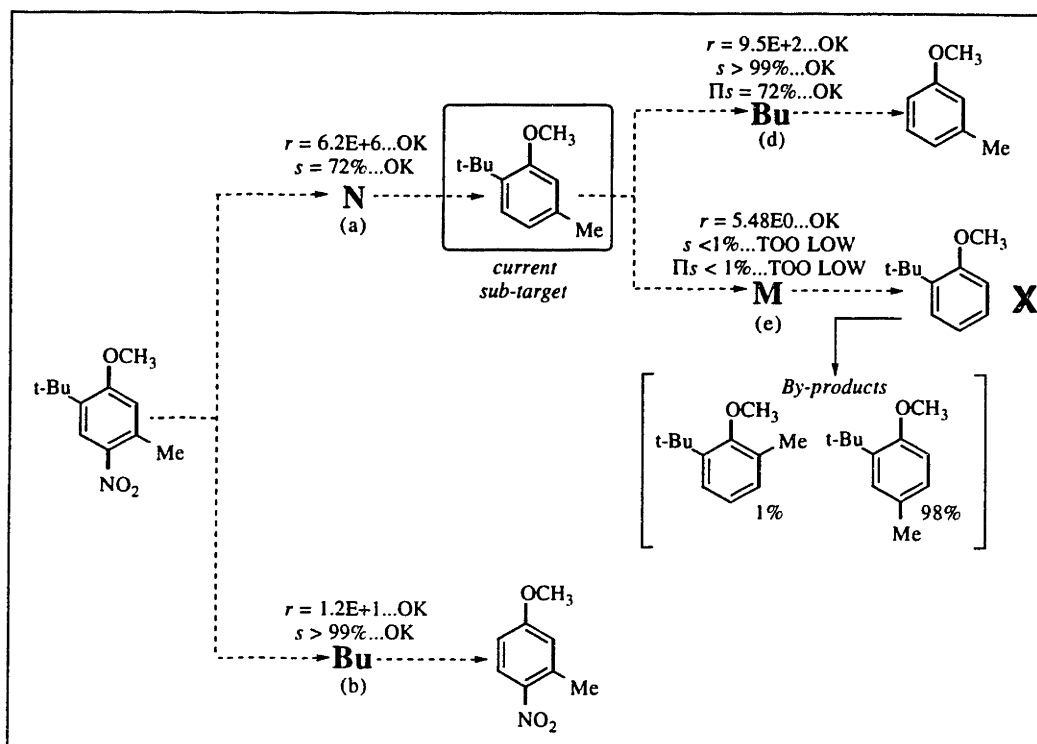


Figure 2.31. Fathoming of 2-tert-Butyl-5-Methylanisole (d-e).

The 3-methylanisole sub-target can be reduced no further due to the reaction rule. Therefore, the analysis branches back up to 2-tert-butyl-5-methylanisole. Fathoming this compound yields two candidate routes (f-g) (Figure 2.32), but the only suitable precursor is again the 3-methylanisole. Since this portion of the reaction network has been explored already, the algorithm terminates.

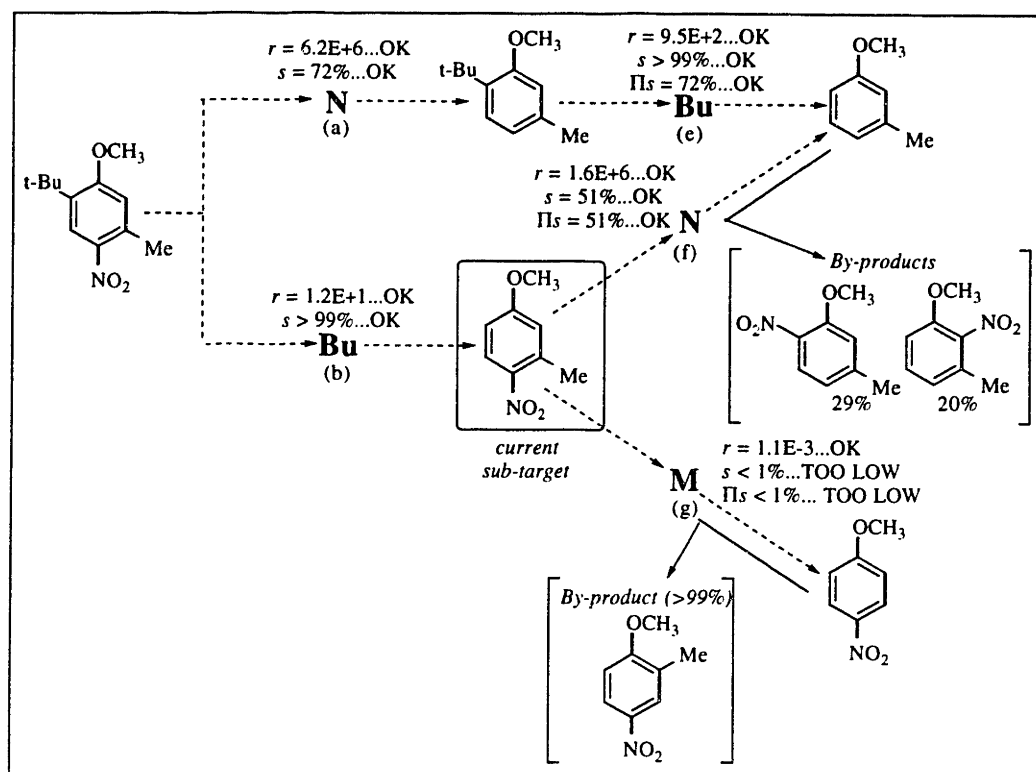


Figure 2.32. Fathoming of 3-Methyl-4-Nitroanisole (f-g). The common 3-methylanisole precursor is identified.

The algorithm produces two candidate reaction paths (Figure 2.33). The route involving butylation followed by nitration is a subpath in a current industrial preparation of musk ambrette (Austin, 1984). The second route switches the reaction order: nitration followed by butylation. Nitration of the butyl-free substrate results in two nitro by-products. The latter route is not desirable by itself, but could potentially exploit synergisms with other reaction paths in the production of related substituted anisole compounds.

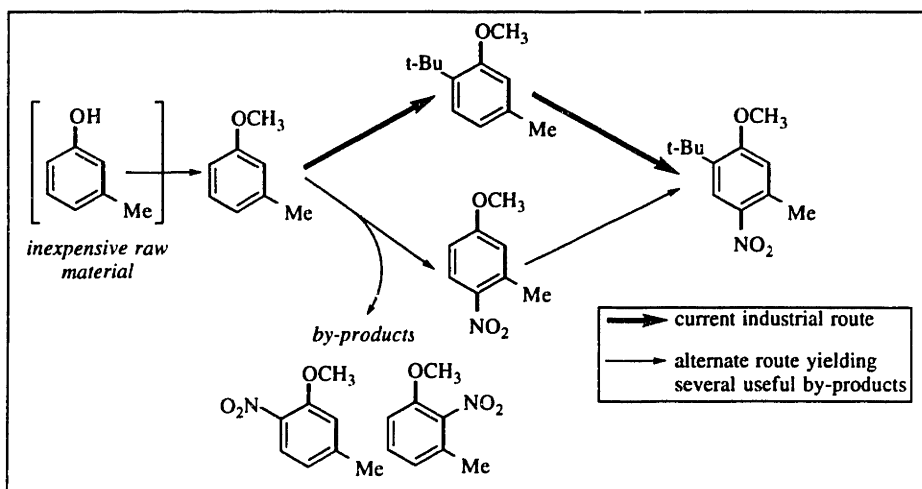


Figure 2.33. Possible Routes to the Target Molecule. Included is a current industrial route (Austin, 1984).

2.9. Discussion

The algorithm developed in this chapter is a kernel for the synthesis of reaction path networks. Whereas the example presented in this chapter is for a single target molecule, a reaction path network generally involves multiple targets. Extension to multiple targets can be accomplished through successive post-processing of the results for single targets. Having generated a reaction path network containing all of the target molecules, topological features can be systematically identified and compared *via* computational analysis. Relevant examples are given in a case study for five benzonitrile compounds in chapter eight.

It should be noted that the algorithm is not restricted to the reaction prediction techniques described above. Based on the assumption of step-wise synthetic organic chemistry, the implicit enumeration framework can be exploited for development of more detailed reaction prediction techniques, reaction rules, as well as molecular property estimation techniques. In principle, calculations related to process elements can also be implemented using the unit process concept (*i.e.*, the set of unit operations corresponding to a generic chemical conversion). The ability to make calculations on both compounds and on individual reactions is a basic theme of this research effort; in particular, the algorithm is viewed as not only a tool for preliminary screening, but also for generating property estimates used in other parts of the research and development effort.

Considering computer-aided design techniques within the set of issues surrounding industrial reaction path synthesis offers a basis for a realistic assessment of the algorithm developed in this chapter. The major limitation of the algorithm is clearly the quality of reaction prediction tools available for extrapolating estimates of reaction rates

from model compound data. The goal of transform network synthesis has been to provide a representation of available chemical transforms that supports "in-house" development of quantitative structure-reactivity models as well as reaction rules. Several comments about such models are in order. First, there has been renewed interest in development of such models in recent years, particularly in the chemistry of certain polymers (McDermott *et al.*, 1990) and hydrocarbon mixtures (Neurock *et al.*, 1989; Nigam and Klein, 1993; Quann and Jaffe, 1992). Second, often the limitations present in "initial" or "unrefined" models can be related to conservative or optimistic assumptions. A good example has been given above in the simplification of a general polysubstitution network to estimate the product selectivity for a reaction. Third, improvements in analysis modules can be tied to new experimental results *via* data assimilation.

2.10. Conclusions

A retrosynthetic analysis algorithm applicable to the synthesis of reaction path networks has been developed. The major components of the algorithm are (i) a recursion scheme for retrosynthetic analysis that facilitates implicit enumeration to minimize computational expense; and (ii) reaction prediction techniques that can be used to address a range of mechanisms encountered in synthetic organic chemistry. Unconstrained application of the algorithm tends to generate an extremely large number of alternatives. To eliminate unsatisfactory paths, quantitative targets and reaction rules based on reaction prediction have been implemented. Preliminary experience in a case study for a synthetic musk indicates that uniform rate and selectivity targets dramatically reduce the combinatorial size of the enumerated reaction paths.

Chapter 3

Integrated Frameworks for Reaction Path Synthesis and Process Development

3.1. Introduction

With the tremendous advances in computing during the past fifteen years, the development of efficient algorithms to solve large, complex design problems has become a major focus of the chemical engineering community. Often the technical scope and limits of the algorithms govern problem representations, mathematical formulations, and even the types of problems considered "important." A basic message in this research is that algorithms are essential, but alone are not enough to systematize process development. As in any complex design activity, the data structures describing chemical engineering design problems are open, dynamic, and subject to the addition and subtraction of dimensions from the design space. A case in point is the evaluation and comparison of reaction path alternatives in process synthesis. For the near future, using any foreseeable techniques, some form of laboratory testing will be essential to the evaluation of new organic chemical reaction paths. Quantitative estimation of the process economics for reaction paths also requires laboratory testing as it is currently not possible to accurately predict whether an arbitrary reaction will work to any degree in the first place and, if it does, levels of by-

products that will be formed. For example, much progress in organic chemistry is resulting from the systematic development and testing of thousands of new catalysts. There are, however, no generally applicable "reaction rules" for catalysts that would lead to a quantitative comparison of two catalysts, either similar or dissimilar in structure.

Some thirteen years after the review by Nishida *et al.* (1981) suggesting self-contained, mathematically rigorous stoichiometric strategies to be the future of reaction path synthesis, essentially the reverse has happened. Most industrial progress has come from the increasing systematization of chemical information, particularly through the use of chemical databases and model building techniques. Several basic conclusions are drawn:

- It is unlikely that the computer can match the creativity of the expert chemist in developing novel chemical conversions, particularly given rapid development new types of chemistry and analytical equipment (*e.g.*, automated micro-scale synthesis) that could revolutionize the laboratory effort.
- In order to make further progress in reaction path synthesis it is necessary to view reaction path synthesis not only as a generation and evaluation problem, but also as an information acquisition problem. An essential goal must be to use a given set of chemical information that is already available on known chemical conversions (*e.g.*, generalized chemical transforms, thermophysical property models, *etc.*) to generate chemical reaction path alternatives so as to yield testable hypotheses in the laboratory. Such an approach provides a link between the discovery of individual chemical conversions in the laboratory and systematic process screening involving integrated configurations of the chemical conversions.
- A more systematic representation of the strategic roles of raw materials, intermediates, and chemical conversions (*e.g.*, the topological dependencies explored in chapter two) are essential to allow laboratory development to rapidly narrow in on the chemical conversions that are feasible as well as industrially promising.

Building upon the current chemical information revolution, a practical and important avenue for reaction path synthesis is the systematic representation of chemical information for the integration of known chemical conversions. Furthermore, it is important to recognize that chemical databases are not the only way in which to codify chemical information. With increased computing power, chemical modeling tools are emerging which combine existing chemical information with model refinement tools. By placing an explicit cost on information, such an approach also motivates the development of formal decision models for experimental design.

This chapter presents a review of industrial reaction path synthesis with an emphasis on developments leading towards information-driven reaction path synthesis frameworks. Following a general introduction to reaction path synthesis, specific

algorithms and strategies are reviewed. The scope of the various approaches are categorized by the following metrics:

- type of chemical information utilized in evaluation;
- integration of generation and evaluation tools;
- capabilities for data assimilation as an integral part of the algorithm;
- scope of the chemistry and generality of the representation;
- incorporation of model building capabilities;
- incorporation of process design criteria;
- use of bounding or pruning strategies to eliminate infeasible reaction paths; and
- ability to exploit increased computing power.

Included in the discussion is an industrial case study undertaken by this author at Polaroid Corporation in which a chemical database is integrated with a retrosynthetic analysis strategy. Finally, several additional developments in model compound experimentation and model building are listed which could play an important role in future efforts.

3.2. Overview

The role of reaction path synthesis in industry is discussed with an emphasis on integration of chemical information tools. A basic conclusion is that an effective computational algorithm must be integrated with an experimental effort. A major role of formal reaction path synthesis is to use available information about individual conversions to identify promising configurations of reaction paths. This role is a critical part of an overall effort, motivating further testing, development, and modeling of design alternatives in an industrial research and development framework.

This chapter reviews industrial reaction path synthesis developments with an emphasis on their generality, choice of evaluation criteria, use of chemical information sources, and incorporation of process-related objectives. Additionally, several reaction prediction formalisms with particular strengths in model building are presented as techniques that could be exploited to improve reaction path synthesis in the future.

3.3. Review of Industrial Reaction Path Synthesis

3.3.1. Background

Reaction path synthesis encompasses a number of strategies concerned with how best to convert a set of raw materials into a set of final products *via* chemical transformations. Reaction path synthesis, broadly defined, is the systematic assembly of generalized

chemical transformations and chemical information to postulate a set of reaction paths possessing desired attributes or satisfying desired targets.

The underlying strategies tend to fall into one of several categories. When the target molecule(s) is specified, a technique known as *retrosynthesis* (Corey, 1969) is often employed, wherein generalized chemical reactions are recursively applied in reverse so as to transform the target molecule into simpler components, ultimately leading to an available starting material for chemical synthesis. The retrosynthesis approach is illustrated for a simple example in Figure 3.1.

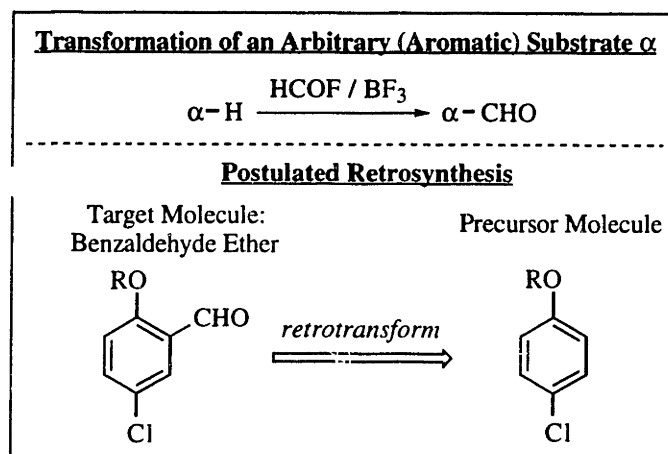


Figure 3.1. Simple Illustration of Retrosynthesis for Formylation by Formyl Fluoride.

In some cases it is desirable to specify the reactant molecule rather than the target molecule. The resulting *synthetic analysis* can facilitate the search for promising product molecules or perhaps screen for undesirable by-products of a proposed chemical reaction. In synthetic analysis the generalized transformations are applied in the synthetic direction and the feasibility of proposed reactions assessed (Salatin and Jorgensen, 1980). In a third situation, partial specification is made of both the target and product molecules and remaining components of the reaction path determined, including by-products, additional reactants, and intermediate transformations. Usually chemical transforms are not expressed explicitly as in the previous two cases but rather in the form of generalized stoichiometric constraints and balances.

Reaction path synthesis techniques such as retrosynthesis were developed as "problem solving techniques for transforming the structure of a synthetic target molecule to a sequence of progressively simple materials along a pathway which ultimately leads to a simple or commercially available starting material for chemical synthesis" (paraphrased from Corey and Cheng, 1989). The success of retrosynthesis and other techniques in the 1970s in generalizing a wide range of chemical reactions, including functional group interconversions (reviewed in Corey and Cheng, 1989) and subtle stereochemical

conversions (e.g., Corey and Long, 1978), quickly catalyzed interest in using reaction path synthesis as a tool in process development. This section reviews the evolution of chemical engineering reaction path synthesis strategies. Assessments are made along the lines described in the introduction.

3.3.2. Original Chemical Engineering Approaches

3.3.2.1. Synthesis of Solvay-Type Clusters

Over the past two decades several approaches using thermodynamic screening criteria have been developed. One of the earliest was for the synthesis of Solvay clusters that satisfy an upper bound on the net change in free-energy (May and Ridd, 1976). A Solvay cluster may be defined as a closed sequence of thermodynamically-favored reactions, each possibly occurring at a different temperature or pressure, that allow a net reaction to proceed that might not be thermodynamically feasible at any single operating condition. The classic example of a Solvay cluster is the original process by Ernest Solvay to convert salt and limestone into soda ash and calcium chloride:

	<u>Reaction</u>	<u>Operating Condition</u>
	$\text{CaCO}_3 = \text{CaO} + \text{CO}_2$	1,000°C
	$\text{CaO} + \text{H}_2\text{O} = \text{Ca(OH)}_2$	100°C
	$\text{Ca(OH)}_2 + 2\text{NH}_4\text{Cl} = \text{CaCl}_2 + 2\text{NH}_3 + 2\text{H}_2\text{O}$	120°C
	$2\text{NH}_3 + 2\text{H}_2\text{O} + 2\text{CO}_2 = 2\text{NH}_4\text{HCO}_3$	60°C
	$2\text{NH}_4\text{HCO}_3 + 2\text{NaCl} = 2\text{NaHCO}_3 + 2\text{NH}_4\text{Cl}$	60°C
	$2\text{NaHCO}_3 = \text{Na}_2\text{CO}_3 + \text{H}_2\text{O} + \text{CO}_2$	200°C
Net Reaction:	$2\text{NaCl} + \text{CaCO}_3 = \text{Na}_2\text{CO}_3 + \text{CaCl}_2$	

May and Ridd (1976) developed a systematic approach for the synthesis of candidate Solvay clusters satisfying stoichiometric balances and the following thermodynamic constraints:

$$\Delta G_r \leq \varepsilon, \quad r \in \text{set of reactions}, \quad (1)$$

where ΔG_r is the free-energy change of a reaction, r , and ε is a positive tolerance that defines a minimum acceptable equilibrium constant of the reaction. An order-of-magnitude value of ε is 10 kcal mol⁻¹, which gives an equilibrium constant *ca.* 0.02 at 1000 °C. Equation 1 establishes heuristic criteria for the feasibility of a set of reactions. These criteria are not sufficient, however, since they ignore the intrinsic kinetics of the reaction, the transport and design considerations, the formation of by-products, and so on.

The key contributions in the work by May and Rudd (1976) rest not in the ultimate feasibility of the proposed mechanisms, but rather in the topological insights that may be gained. They have shown that any sequence of reactions in a Solvay cluster must fall into one of four general categories: (i) reactants in the net reaction, denoted A, B, C, \dots ; (ii) products in the main reaction, denoted Z, Y, X, \dots ; (iii) L -class intermediates that serve as reactants in a Solvay cluster, denoted L, K, J, \dots ; and (iv) N -class intermediates that are generated by the reactions involving L -class intermediates, denoted N, O, P, \dots . As an example, the general reaction



might have the following Solvay cluster:



May and Rudd (1976) developed a general graph representation of such reactions. Species are represented by arcs connected such that any closed cycle in the graph is a reaction or linear combination of reactions. In the case of $A + B = Z$, the Solvay cluster may be represented as shown in Figure 3.2.

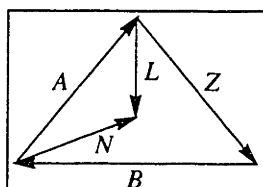


Figure 3.2. Graph Representation of a Solvay Cluster.

As can be seen from the graph, all of the L and N species are in the form of nested polygons within the outer polygon formed by the net reactants and products. The presence of invariants of the graph is revealed by the fact that the arrows can be arranged consistently so that in all cycles contain reactants in a clockwise direction and all products in a counterclockwise direction; thus each cycle in the graph must be acyclic. In particular, any sequence of arcs forming the boundary of a nested polygon must not all follow the same direction.

The synthesis procedure developed by May and Rudd (1976) consists of starting with a minimal polygon representing the net reaction (*e.g.*, the triangle ABC) and recursively introducing nested polygons involving new species that satisfy the topological requirements of the graph. The choice of species is based on the selection of a common species or a set of species that participate in a thermodynamically favorable reaction. A

thermodynamically feasible candidate results when all of the interior cycles satisfy Eq. 1. Graphically, the procedure consists of choosing an arc or a set of arcs and introducing an internal polygon connected to the arc(s) that represents a new reaction. In the case of the graph shown in Figure 3.2, the arc A has been chosen and the species L and M introduced, thereby partitioning the net reaction into the two reactions forming a Solvay cluster.

3.3.2.2. Generalized Stoichiometry

A second systematic approach to the synthesis of stoichiometric reaction paths that also exploits temperature shifts to control the equilibrium constant has been developed by Fornari *et al.* (1989). Their method is applicable to industrial gases and other light molecules. The genesis for the method is the generalized stoichiometry formula, given by the under-determined system of equations in Eq. 5.

$$E \cdot \nu = 0, \quad (5)$$

where E is the $[e \times (s+1)]$ -dimensional matrix of atomic coefficients and ν is the $(s+1)$ -dimensional vector of stoichiometric coefficients, where it is assumed that $e < s$. The term e is the total number of atomic elements. The term $(s+1)$ is the total number of species including the product (the quantity s is convenient to work with since the product is a given and is assigned a stoichiometric coefficient of unity).

Defining $m \equiv s - e$ as the dimension of the null space in Eq. 5, Fornari *et al.* (1989) studied the thermodynamic feasibility of chemical reactions for case of $m = 0, 1$, and 2 . As in the case of May *et al.* (1976), the reaction screening criteria is the thermodynamic constraint (Eq. 1). A key assumption in the approach is that the standard free-energy of reaction is a linear function of the temperature, T :

$$\Delta G = (\alpha^t + \beta^t T) \cdot \nu. \quad (6)$$

From Eqs. 1 and 6 and taking into account the number of degrees of freedom, m , Fornari *et al.* (1989) show that various classes of reactions either fall on straight lines or intersect at different points in the $(\Delta G, T)$ region. As an example, consider the case $m = 2$. Specifying $m = 2$ means that the vector ν can be written as Eq. 7.

$$\nu = (v_1, v_2, \dots, v_{s-2}, q_{s-1}, q_s, 1), \quad (7)$$

where specifying the stoichiometric coefficients q_{s-1} and q_s determines the remaining coefficients *via* Eq. 5. As part of their analysis, Fornari *et al.* (1989) proved the following four properties (P1-P4):

- P1: Given common $q_{s-1} = q_s = c$, $c \in (-\infty, \infty)$, all reactions intersect at a common, well-defined point $(\Delta G_{s,s-1}, T_{s,s-1})$.
- P2: The common abscissa, $T_{s,s-1}$, is independent of the composition of the product (as long as no new elements are introduced).
- P3-P4: All lines representing reactions for which $q_{s-1} = c_{s-1}$ intersect at a common point $(\Delta G_{s-1}, T_{s-1})$ and all lines representing reactions for which $q_s = c_s$ intersect at a common point $(\Delta G_s, T_s)$. Furthermore, in each case the temperature corresponding to the locus of points is independent of the product and c_{s-1} or c_s , respectively.

It follows from Properties 1-4 that all reactions $q_{s-1} = q_s = c$ fall on a single line defined by any two of the points $(\Delta G_{s,s-1}, T_{s,s-1})$, $(\Delta G_{s-1}, T_{s-1})$, and $(\Delta G_s, T_s)$. Furthermore, it is shown in an additional property that the points $(\Delta G_s, T_s)$ and $(\Delta G_{s-1}, T_{s-1})$ belong to the line corresponding created by $q_s = c_s$ and $q_{s-1} = c_{s-1}$. As a final note, the analysis has been extended to show how the line shifts in a predictable way to changes in species.

The topological invariants discussed above have been exploited to develop bounds on the range of feasible stoichiometric coefficients and temperatures satisfying the thermodynamic criteria. Using the bounds to prune infeasible paths from consideration, a computerized search procedure has been developed to synthesize classes of overall reactions producing a desired product. The inputs to the method are the number of species, $(s+1)$, the identity of possible species and their categorization as raw materials or products, the number of elements, and the parameter vectors, α and β , for computing the free energy.

3.3.2.3. Electron-Pushing Schemes

Whereas the previous two approaches are for reactions involving relatively small molecules in inorganic chemistry and industrial gas chemistry, a key area of current interest is obviously the chemistry of larger organic compounds. The applicability of generalized stoichiometry arguments for proposing new organic synthetic routes is significantly limited by (i) the interdependence between reaction pathways and structural attributes of molecules; and (ii) the fact that simple thermodynamic screening criteria do not provide strong bounds for screening organic chemical reactions. For example, an accurate measure of free energy would provide little insight as to whether one might accomplish the unlikely reaction depicted in Figure 3.3.

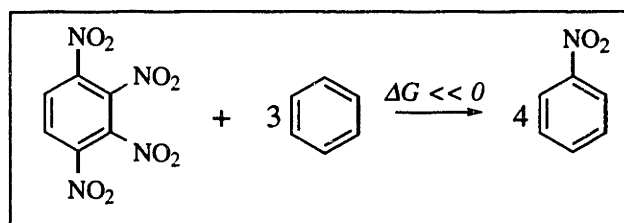


Figure 3.3. Mechanistically Infeasible Reaction That Could Satisfy a Free Energy Screening Criterion.

In general, use of generalized stoichiometry-based methods on organic molecules yields an extremely large number of alternatives involving wild permutations of the molecular structures of the reactant molecules.

Agnihotri *et al.* (1980) argue that an explicit treatment of molecular structure is desirable in reaction path synthesis. In their view of research and development, one is presented with a set of strategic raw materials to be converted into economically valuable products. Therefore, for a particular set of chemical reactions, one would like to know what products are likely to be produced. As shown in Figure 3.4, Agnihotri *et al.* (1980) envision an approach to research and development where (i) the spectrum of likely products is identified through reaction path synthesis; and (ii) economically valuable products are promoted through reaction engineering such as optimization of operating conditions and choice of catalyst.

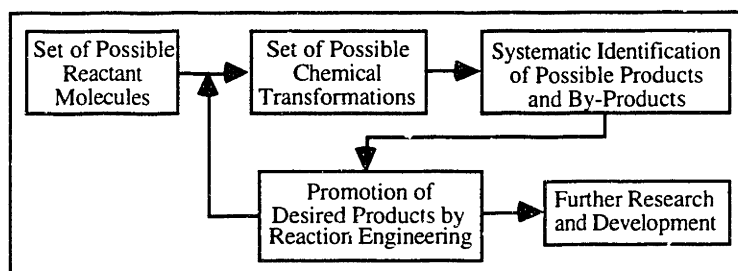


Figure 3.4. Interpretation of the Scheme Envisioned by Agnihotri *et al.* (1980) for Improved Industrial Research and Development.

The first step in the approach of Agnihotree *et al.* (1980) is the representation of molecular structures in terms of molecular bonds that can react to form a product molecule. Agnihotri *et al.* (1980) represent molecular structures with a connectivity matrix between atoms in which the magnitude of the matrix elements indicate the number of electrons shared between two atoms. For example, Figure 3.5 shows the connectivity matrix for α -hydroxy acetonitrile.

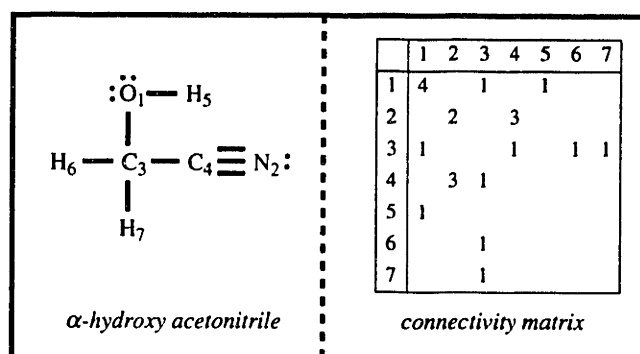


Figure 3.5. Illustration of the Connectivity Matrix Representation. Diagonal entries represent the number of unshared valence electrons, off-diagonal entries represent the number of bonding shared electrons between two atoms. Adapted from Agnihotri, et al. (1980).

The entries in the connectivity matrix are the number of valence electrons that are shared between two atoms. Because the electrons are assumed to be shared equally, the matrix is symmetric. Diagonal entries represent the number of unshared electrons on an atom.

The relationship between the electron distribution and the atomic connectivity thus established, a chemical reaction is defined as the matrix difference between the reactant and the product molecule. Letting B be the starting connectivity matrix, R the reaction matrix, and E the final connectivity matrix, the following equation holds:

$$R = E - B \quad (8)$$

The reactivity matrix essentially pushes electrons around the set of atoms to achieve a new connectivity and bonding order. To illustrate, Figure 3.6 shows the decomposition of α -hydroxy acetonitrile into formaldehyde and hydrogen cyanide.

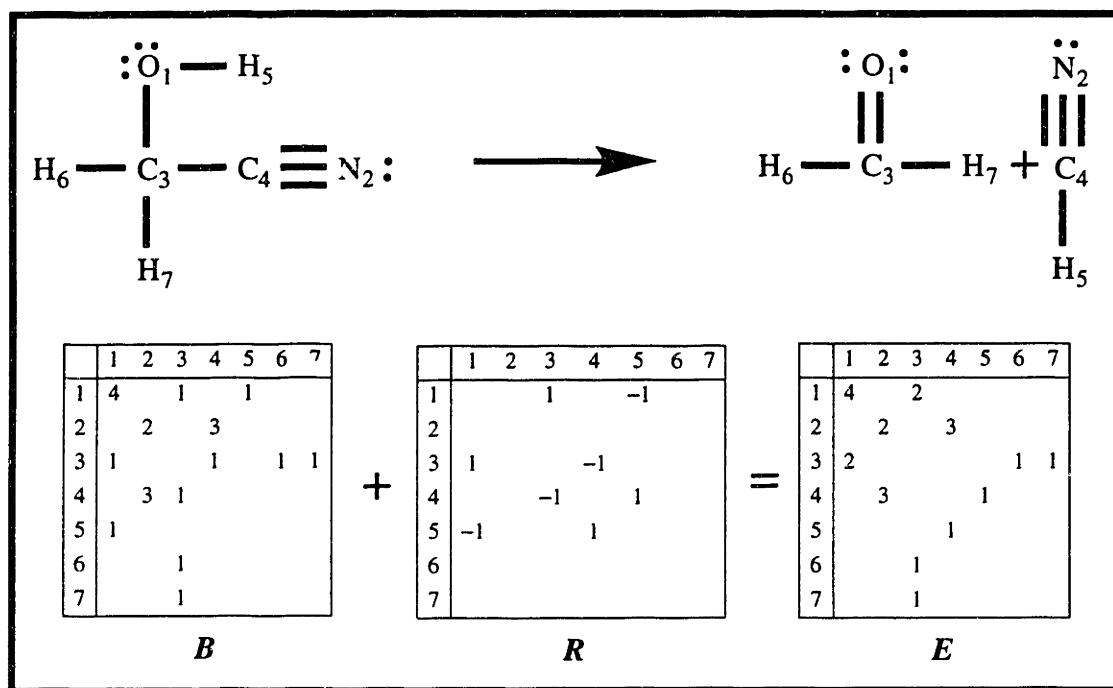


Figure 3.6. Illustration of the Definition of a Reactivity Matrix, R , in Terms of the Reactants, B , and the Products, E , That Are Formed.

The main idea embodied in an additive transformation such as Eq. 8 is that a class of chemical reactions with certain matrix invariants can be used to generalize the notion of isomerism to include the interconversion of reactant and product molecules. In other words, for a fixed set of atoms, all of the bond patterns that can be generated by a class of R -matrices give rise to molecules that are considered isomers of one another. Furthermore, the molecules that result from the recursive application of the R -matrices are also isomers.

The set of R -matrices that have been considered include those that possess the same bond redistribution pattern applied to different atoms in the reactant molecule. The set of such matrices is termed an R -category. All of the matrices in an R -category can be generated by an "irreducible R -matrix." In the work presented by Agnihotri *et al.* (1980), consideration was given to a class of so-called restricted chemistry in which several constraints are imposed on the R -matrices:

- the R -matrix is symmetric [and integral] by construction;
- the diagonal elements are zero—*i.e.*, unshared electrons are not redistributed; and
- the sum of every row is zero and the sum of every column is zero.

A basic consequence of the constraints is that no charge separation is evolved during a reaction. It should be noted that the symmetry requirement is needed whenever a

symmetric representation of the molecule matrix is used. It can be shown that the number of R -matrices in an R -category satisfying the constraints is

$$\frac{n \times (n - 3)}{2}, \quad (9)$$

where n is the number of atoms in the molecule whose bonds are to undergo rearrangement (*i.e.*, n is the dimension of the irreducible R -matrix), and that the minimum number of such atoms is 4.

Additional constraints can be derived for restricted chemistry that specify the sets of atoms allowed in an ensemble generated by an R -category. The constraints are a function of several characteristic descriptors of the reactions:

M : number of bonds made and broken;

N : number of atoms involved in the reaction; and

i : subcategory index (if more than one category exists for a choice of N and M).

The number of bonds, M , is equal to the sum over all monovalent, divalent, *etc.* atoms (Eq. 10).

$$M = \frac{1}{2} \sum_{j=1}^r j S_j, \quad (10)$$

where S_j is the total number of j -valent atoms. Furthermore, the total number of atoms, N , is given by Eq. 11.

$$N = \sum_{j=1}^r S_j. \quad (11)$$

Constraining the admissible valencies for a given choice of M and N , an additional invariant of an R -category can be derived by subtracting Eqs 10 and 11 to yield Eq. 12.

$$2M - N = \sum_{j=2}^r (j - 1) S_j. \quad (12)$$

Equation 12 provides some insight into the admissible atom set. In particular, Agnihotri *et al.* (1980) state that most of the interesting chemical reactions are limited to relatively small values of no more than M and N *ca.* 4 and 6. For example, consider the case of $M = 2$ and $N = 4$; Equation 10 implies that all four atoms participating in the

reactions are monovalent. In other words, the only possible configurations available for atoms A, B, C, and D are A—B and C—D; A—C and B—D; and A—D and B—C.

The possible reactions that can be generated consist of different bond reorganization patterns between the four atoms. All such reactions can be generated by permuting the rows and columns in any one *R*-matrix elicited from the *R*-category. For example, consider the *R*-matrix in figure 3.7.

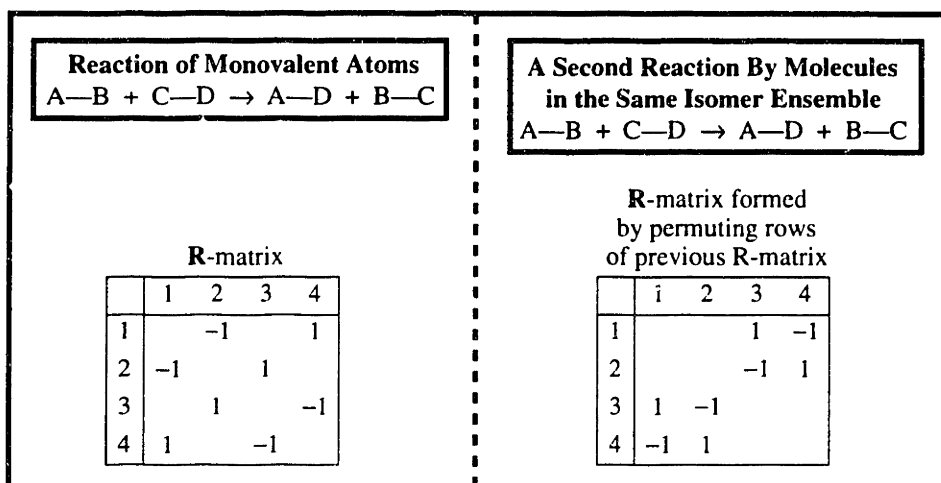
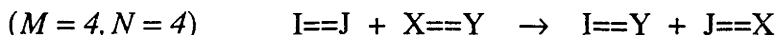
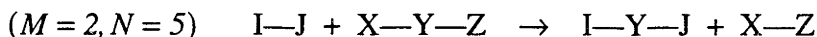
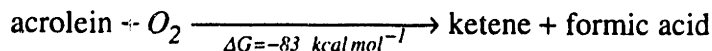
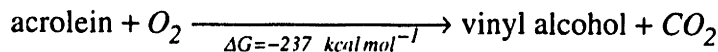
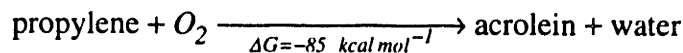
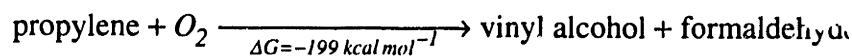


Figure 3.7. Illustration of Different Reaction Matrices Within a Single *R*-Category That Can Be Generated for Different Bond Redistribution Patterns.

Furthermore, by inserting rows and columns of zeros into the *R*-matrix, a molecule of arbitrary size can be handled so long as the atoms undergoing reaction (*i.e.*, non-zero rows and columns in the *R*-matrix) obey the atom constraints of the *R*-category. As a result, atomic elements can be specified so that the *R*-categories correspond to industrially relevant reaction classes. Other examples of *R*-categories include:



In their implementation of the bond reorganization scheme, Agnihotri *et al.* (1980) specify the initial reactants and the set of available reactions and use them to generate intermediates, which are then re-reacted repeatedly to form product molecules. The method has been illustrated for the oxidation of propene at a specified temperature for the *R*-categories ($M = 4, N = 4$) and ($M = 2, N = 5, i = 1$). The screening criterion used to select potentially feasible routes is thermodynamic in nature, as in the previous examples. For ($M = 2, N = 5, i = 1$), the thermodynamically favored routes ($\Delta G \leq 0$) are:



3.3.2.4. Retrosynthetic Analysis

Another contribution to reaction path synthesis that exploits a knowledge of molecular structure and reaction specificity is that of Govind *et al.* (1981). Theirs is an interesting example to consider because it offers a view of reaction path synthesis that contrasts to some extent with that of Agnihotri *et al.* (1980). It should be recalled that Agnihotri *et al.* (1980) explore the space of possible reaction paths in a synthetic direction—*i.e.*, starting from a set of strategic raw materials and working forward to identify new products. They argue that by identifying the alternate outcomes of a reaction(s), it should be possible to use reaction engineering to find conditions that favor the selectivity of the most desirable routes. Govind *et al.* (1981) argue that a major goal of reaction path synthesis should be to start with target products, such as important industrial organic chemicals, and to work retrosynthetically—*i.e.*, backwards—to identify new raw materials. The key distinction between the two research approaches is that the former is willing to consider new types of reactions while fixing the set of raw materials, whereas the latter uses existing chemical reactions in different permutations to identify new raw materials.

The main application envisioned by Govind *et al.* (1981) was the development of new feedstocks to replace then skyrocketing petroleum-based feedstocks. While this motivation is somewhat outdated (at least temporarily), the perceived need to rethink strategic selection of raw materials and intermediates is relevant to modern environmental concerns. In particular, Govind *et al.* (1981) identified several characteristics of reaction paths in the commodity and specialty chemicals industries that are still relevant today in considering new strategies for reaction path synthesis:

- Target molecules (*i.e.*, products) are relatively small by the standards of modern synthetic organic chemistry. Most products consist of fewer than twenty heavy atoms.
- Due to the fact that most commodity and specialty chemicals ultimately come from petroleum, their production schemes are interconnected by a network of raw materials, intermediates, and products.

- At a preliminary stage of economic evaluation, it is highly desirable to know the set of by-products, the overall stoichiometry, and the yield.
- In the modification of an existing production route, characterizing impurity reactions is extremely important to avoid loss in product performance and market value.

Govind *et al.* (1981) proposed a method to generate alternative reaction paths that in principle could be extended provide a preliminary economic evaluation. The essential tool used by Govind *et al.* (1981) to search for new sources of raw materials is retrosynthesis analysis augmented by structural reaction rules. Such a procedure can be applied recursively, resulting in a network of molecules connected by arcs representing different chemical transformations. An example of a retrosynthetic transformation and its application to a target molecule is illustrated in Figure 3.8.

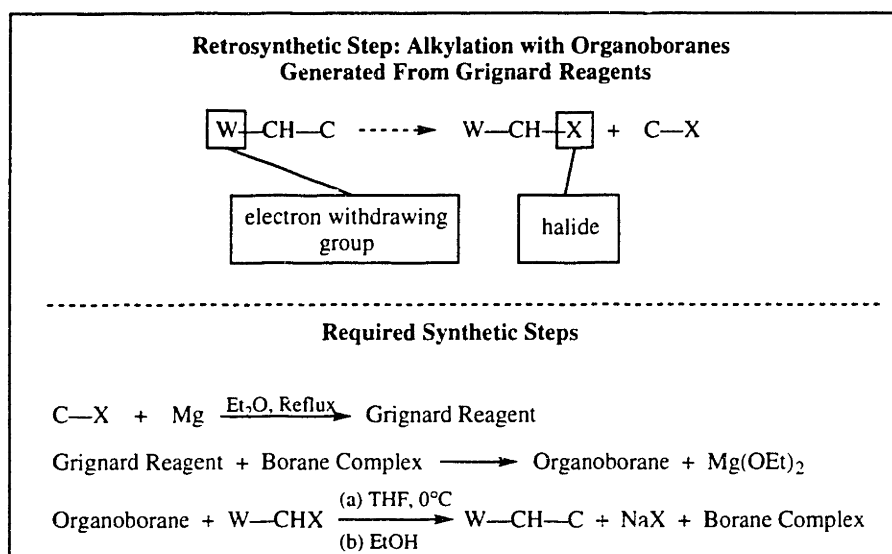


Figure 3.8. Illustration of a Retrosynthetic Transformation Applied to a Substructure of a Molecule and the Corresponding Synthetic Transformations That Are Required. Adapted from Govind, *et al.* (1981).

As can be seen in Figure 3.8, a retrosynthetic transformation operates on a substructure of a molecule to generate a new substructure, giving rise to a precursor molecule. Consequently, a retrosynthetic transform can potentially be applied to many different molecules that share the structural features salient to the chemical transformation. One of the major challenges in using retrosynthetic transforms, however, is to determine whether competing functional groups or other molecular substructures will interfere with the desired transformations. In borane-catalyzed alkylation, for example, the retro-reaction probably does not apply if the two carbons in the product molecule are part of a common ring structure. Thus, an effective approach to retrosynthetic analysis must combine the use

of retrosynthetic transformations with an analysis of all the relevant molecular substructures in both the target and precursor molecules.

Given a target molecule, Govind *et al.* (1981) propose an automated approach to retrosynthetic analysis that (i) identifies possible applications of retrosynthetic transformations and (ii) subjects them to feasibility constraints based on logical reaction rules. They have constructed a list of approximately two hundred industrially useful retrosynthetic transformations with the accompanying synthetic conversions. In addition, approximate operating conditions and a list of literature references are included.

The first step in the approach due to Govind *et al.* (1981) is the identification of candidate retrosynthetic transformations. In general, several retrosynthetic transformations may be applicable to a target molecule and perhaps a single transformation may be applicable at several locations on the target molecule. Similar transformations may also be applicable at different locations on the synthetic tree. A portion of the synthetic tree to methyl cysanthemate serves to illustrate (Figure 3.9).

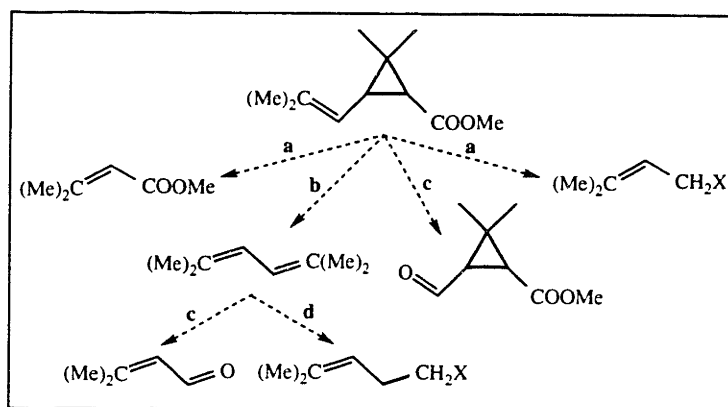


Figure 3.9. A Portion of the Candidate Transformations That Can be Generated for Methyl Cysanthemate. The individual transforms are denoted by a-d. a corresponds to cyclopropanes via sulfur ylides; b corresponds to cyclopropane via diazoacetate intermediate; c corresponds to thermal coupling of organocuprates; and d corresponds to the Wittig reaction.

In the computational implementation, both molecules and transformations are represented by a connectivity matrix similar to that used by Agnihotree *et al.* (1981). In the work of Govind *et al.* (1981), however, the relevant molecule substructures must be "perceived" and the generalized transform manipulated into the appropriate context.

In order to assess the feasibility of the candidate transformations, Govind *et al.* (1981) propose a second step consisting of logical checks that must be satisfied before the retrosynthetic transformation can be considered for use in a chemical reaction path. For example, in the case of alkylation of organoboranes generated from Grignard reagents, the following reaction rules are proposed (the rules refer to Figure 3.8):

In the substructure $W-C_1-C_2-C_3$,

- kill (*i.e.*, eliminate the candidate conversion from consideration) if W is not an ester or a nitrile;
- kill if there is no hydrogen attached to C_1 , C_2 , or C_3 ;
- kill if C_1 is part of any functional group and not an alkyl halide;
- kill if C_2 is part of any functional group;
- kill if bond (C_1-C_2) is part of a ring substructure;
- if C_3 is part of any functional group, then pass C_3 if it is an ether or tertiary amine, otherwise kill;
- kill if there is a heteroatom attached to a carbon adjacent to C_3 ;
- *etc.*

Govind *et al.* (1981) implemented a set of thirty-eight generic logical functions that can be assembled in different combinations to form reaction rules such as those given above. The resulting rules are applied serially to the molecule under consideration. The codification of chemical logic using generic logical functions has several inherent advantages. Once the logic is implemented, it can be ported to a database, extended to a higher level of detail as new experimental results become available, and accessed in the future. For example, the information might be accessed under different keywords, including the type of molecular structure, the type of chemical reaction, and operating conditions.

For those retrosynthetic transformations that satisfy the constraints generated by available reaction rules, a third level of evaluation is available in which the stability of the targets and precursors is tested for the particular choice of reagents. The test consists of comparing the set of functional groups present on the target molecule with a table listing which functional groups are unstable in the presence of different reagents.

Several applications have been published by Govind *et al.* (1981), including synthesis routes to methyl chrysanthemate and ϵ -caprolactam (Govind *et al.*, 1976; Govind, 1977). Portions of the routes to methyl chrysanthemate are given in Figure 3.9. In particular, identified routes to the cyclopropane ring are similar to several others proposed independently for chrysanthemic acid esters.

3.4. Extensions of Reaction Path Synthesis of Relevance to Chemical Engineering

3.4.1. Chemical Modeling in Reaction Path Synthesis

One of the first papers to consider chemical modeling as an integral part of reaction path synthesis is "Computer-Assisted Reaction Prediction and Synthesis Design," by Gasteiger *et al.* (1990). In their software for retrosynthetic analysis, named EROS 6.0, they have implemented empirical techniques for the estimation of physicochemical properties. The properties include

- partial atomic charges of atoms;
- inductive, resonance, and polarizability of molecular bonds;
- heats of reaction; and
- bond dissociation energies.

Such information is implemented in multivariate statistical correlations designed to predict the outcome of the retrosynthetically-generated candidate reactions (an example is given below).

In principle, their use of quantitative information allows for much more formal integration of reaction path synthesis *and* evaluation than previously considered. Gasteiger *et al.* (1990) begin by pointing out that the problem of designing a synthesis route by retrosynthetic analysis is closely linked to reaction prediction—*i.e.*, the process of determining whether a precursor will indeed react to form a target molecule. They propose to combine synthesis design and reaction prediction by systematically testing retrosynthetically-generated precursors for the desired reactivity. An outline of the approach is given in Figure 3.10.

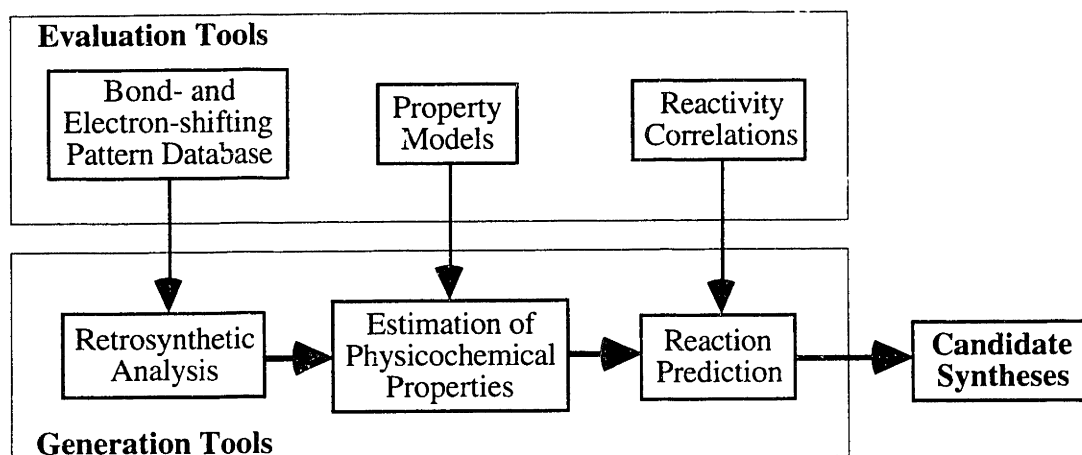


Figure 3.10. Outline of the Combined Retrosynthetic Analysis–Reaction Prediction Strategy of Gasteiger *et al.* (1990) That Exploits the Automated Generation of Physicochemical Properties.

A hypothesis underlying their approach is that property prediction models can be exploited to avoid narrowly defined chemical transforms in favor of generalized reactions consisting of patterns of reorganized bonds. Such reactions can potentially be applied to many different instances of functional groups and molecular structures, potentially leading to "new" chemistry. Consequently, the initial retrosynthetic step is not based on transform selection but rather on a bond- and electron-shifting strategy. In reaction prediction, alternate bond-shifting patterns are compared *via* the physicochemical correlations to identify the most likely pattern of reorganized bonds for a chemical reaction.

The methodology of Gasteiger *et al.* (1990) has been illustrated for the simple problem of synthesis of *meta*- and *p*-chloronitrobenzene by electrophilic aromatic substitution. In EROS 6.0, Gasteiger *et al.* (1990) adopt the following high-level strategies in order to systematically identify a synthesis route:

- remove each substituent to generate a precursor molecule, starting with the most electronegative substituent; and
- test each precursor to determine whether the substituent will be reintroduced at the desired position.

Although the particular problem they consider can be solved by inspection through elementary rules of orientation (*i.e.*, chlorinate nitrobenzene to achieve *meta* orientation; nitrate chlorobenzene to achieve *para* orientation), it serves to illustrate the quantitative measures they propose for reaction prediction. In the case of electrophilic aromatic substitution, three separate descriptors of reactivity and selectivity are employed:

- (i) the degree of electron withdrawal, W , of a substituent leading to polarization of a σ -bond;

- (ii) the activation, A , of a substituent due to resonance; and
- (iii) the selectivity, S , of open positions on the ring due to a substituent.

As shown in Table 3.1, the magnitudes and signs of W , A , and S reproduce the textbook rules of orientation.

Table 3.1. Computed Parameters for Reaction Prediction *p*- and *m*-Chloronitrobenzene.

Position	Reactivity Measures		
	W	A	S
<i>p</i> -Chloronitrobenzene			
1-Cl	1.58	-0.36	-0.18
4-NO ₂	12.26	0.55	0.28
<i>m</i> -Chloronitrobenzene			
1-Cl	1.57	0.0	0.18
3-NO ₂	12.26	0.0	-0.28

In the case of the para-substituted compound, the values of the descriptor W show that the nitro group is the more electronegative substituent. The positive values of the descriptors A and S indicate that para substitution of the nitro group is favored, although no calculations are presented for the ortho position. Similarly, the descriptors can be used to evaluate the most probable synthesis of the meta-substituted product. Again, the nitro group is the more electronegative. The selectivity descriptor, S , of the meta position, however indicates that substitution there is not favored. In the case of meta-directed chlorine substitution, however, the descriptor S indicates favorable chemistry.

3.4.2. DNA Synthesis Routes

An example that involves reaction path screening based on process design objectives has been considered by Powers *et al.* (1975) in the optimization of synthesis routes to bihelical DNA (deoxyribonucleic acids). Since the driving cost in the synthesis of new DNA is the laboratory time involved in reaction, separation, and analysis (raw material costs are small in comparison), the optimization objective is to minimize the synthesis time. In order to estimate the time accurately, Powers *et al.* (1975) perform in-depth modeling of the reaction, separation, and analysis requirements for different synthesis paths. In particular, they account for product yields, molecular weight dependencies in the separations times, reagent costs, and other factors—all information that is very useful to the researcher about to implement the results of the analysis.

DNA molecules consist of assemblies constructed from four different nucleotide bases: Adenine (A); Guanine (G); Cytosine (C); and Thymine (T). A convenient notation

for DNA molecules is to represent the individual helices as a sequence of the bases A, G, C, and T. Figure 3.11 illustrates a segment from a DNA molecule:



Figure 3.11. Typical Segment of Bihelical DNA. The segment consists of two strands: ATTGC and TAACG.

As can be seen in Figure 3.11, the bases A and T or G and C always associate between two strands. Furthermore, given one of the strands, it turns out that the other strand can be synthesized by spontaneous self-assembly in a solution containing one strand and a mixture of the individual acids.

The problem addressed by Powers *et al.* (1975) is to identify the optimal synthesis routes to one of the strands in a target bihelical DNA molecule. The first part of the approach consists of an algorithm for the systematic generation of alternative syntheses that exploit bounds obtained from an evaluation function. A synthetic step can be defined as the addition of two smaller segments of a DNA strand to form a larger DNA strand. The set of possible reactions can be generated systematically by partitioning a target molecule into different segments. For example, the molecule ATTGC can potentially be formed by four different reactions:

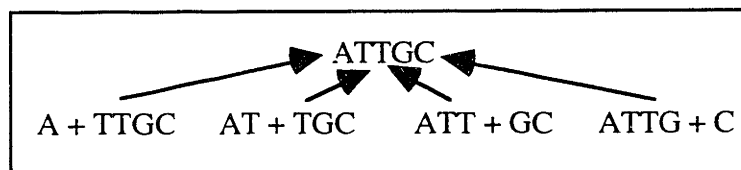


Figure 3.12. Possible Combinations of Raw Materials Used in the Synthesis of a DNA Segment.

By applying the procedure recursively, the set of all synthesis routes terminating in the single acids could in principle be generated. As a result, it is possible to use retrosynthetic analysis to generate all of the synthesis routes. In the case of DNA synthesis, Powers *et al.* (1975) report that stronger bounds are available at the beginning of the synthesis route—*i.e.*, when the molecular weights of the reacting compounds are smaller. In the development of any combinatorial optimization procedure where bounding information can be extracted, it is usually desirable to search the combinatorial space in such a way that exploits the strongest bounds first, since stronger bounds allow non-optimal designs to be recognized and discarded with minimum effort. As a result, it is desirable to construct synthesis routes starting with the raw materials—A, T, C, G—and proceed through the network of intermediate molecules towards the product molecule. One of the traditional problems associated with such an approach is guaranteeing the

convergence of different paths to the same product molecule. Due to the simple linear structure of the DNA assembly, however, the set of intermediate molecules can be identified *a priori* as the power set of subsets of DNA strands from the product molecule. Exploiting the up-front knowledge of the set of precursors, Powers *et al.* (1975) propose a convergent dynamic programming strategy for the generation and evaluation of the synthesis routes (Figure 3.13).

ATTGC A+TTGC AT+TGC ATT+GC ATTG+C			
ATTG A+TTG AT+TG ATT+G		TTGC T+TGC TT+GC TTG+C	
ATT A+TT AT+T	TTG TT+G T+TG	TGC TG+C T+GC	
AT A+T	TT T+T	TG T+G	GC G+C
A Start	T Start	G Start	C Start

Figure 3.13. Illustration of the Procedure for Building Up a DNA Fragment from Smaller Fragments. The fragments in bold illustrate the intermediates; all of the intermediates can be synthesized from smaller fragments evaluated previously.

An important feature of the problem that makes a forward, or synthetic, approach suitable is the fact that there are only four possible raw materials. In the case of the reactions depicted in Figure 3.13, a cost and time is initially computed for A+T, T+T, T+G, and G+C. The bounding procedure begins by considering trinucleotides such as **TTG**. The combinations of a mono- and dinucleotide that result in the smallest synthesis times for trinucleotides are chosen and the remaining paths eliminated. A similar procedure is adopted for tetra- and higher substituted DNA fragments. In each case, the goal is to find the synthesis step that results in the minimum time being expended to make a given DNA segment. The most efficient synthesis route is guaranteed at each size of DNA fragment, including the largest, which corresponds to the product molecule. The efficiency of the method comes from the fact that synthesis paths that contain inferior subpaths are systematically eliminated from consideration.

Underlying the approach presented above is the need to estimate time, yield, and cost estimates for each synthesis step. The reaction procedure for combining two DNA fragments is a standard technique consisting of (i) protection of appropriate amino, hydroxyl, phosphate, and oligonucleotide functions present on the two DNA strands; and (ii) repeated condensations of the two protected DNA strands to form the product. The number of protection reactions and the time required for each reaction are determined by the analysis. Powers *et al.* (1976) proposed an empirical power-law dependence for the attention time, T , required for protection, condensation, and duplex joining reactions as a function of the number of grams of reactant, Q :

$$T = T_0 \cdot (Q/Q_0)^{0.3}, \quad (13)$$

where T_0 is the observed attention time at a given quantity of reactant Q_0 . Equation 13 implicitly includes the effect of yield losses because such losses require that a greater quantity of reactant, Q , be used to make a specified amount of product. Powers *et al.* (1976) quantify the yield of a product with an empirical function that depends on several features of the structure of the target molecule.

In the cases of separation and analysis in each synthetic step, Powers *et al.* (1976) develop similar evaluation criteria. The separations proceed *via* ion exchange chromatography. An appropriate model of the total separation time is developed that accounts for the charge differences between the products, by-products, and raw materials present in the reaction mixture.

The optimization algorithm has been applied to a number of DNA syntheses. In one case the synthesis route to a gene has been reduced from 22 man-years to 11 man-years. Although the research Powers *et al.* (1976) was done nearly two decades ago, there has been renewed interest in the approach recently in connection with automated laboratory robots. In addition, advances such as polymerase chain reaction make it desirable to synthesize larger DNA molecules which can then be produced in abundance.

3.5. Future Directions of Reaction Path Synthesis

3.5.1. Model Building: Stochastic Kinetics

There is a continuing need to show how reaction path synthesis can really fit into research and development. A key part must be the assimilation of experimental data and model building. McDermott *et al.* (1990) present an integrated approach to data assimilation and

mechanistic modeling based on a systematic representation of the kinetic and thermodynamic "building blocks" for lumping intrinsic reaction pathways, thermodynamic effects, and transport phenomena. As shown in Figure 3.14, such models are parameterized in part by model compound experiments that are developed for the specific systems of interest. The assimilation of model compound experiments used to identify the intrinsic reaction paths in the system of interest is a major contribution of the work since it allows the intrinsic kinetics of reactions occurring on a complex molecule to be determined on smaller, more easily studied substrates. Such an approach is valuable in process scaleup if transport phenomena mask the intrinsic kinetics of a reaction.

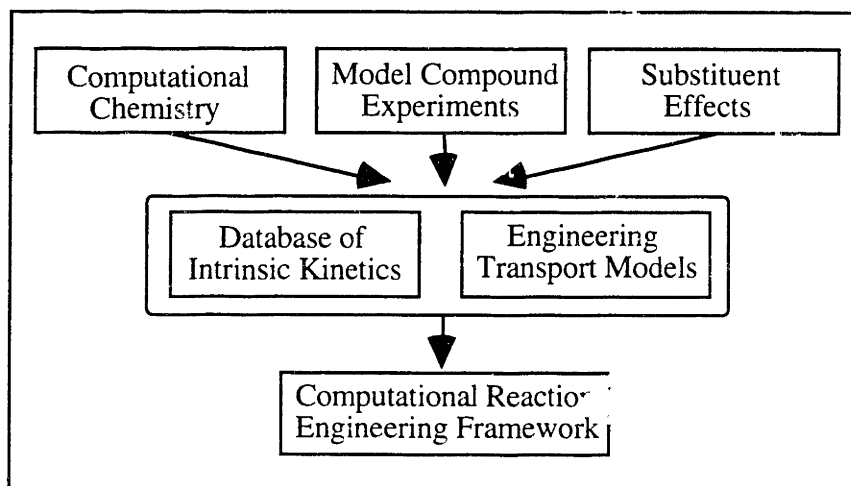


Figure 3.14. Chemical Modeling Formalism Being Developed by McDermott et al. (1990).

The methodology has been illustrated in a number of different reacting systems. Discussed here is the simulation of the depolymerization (liquefaction) of polyveratryl β -guaiacyl ether (polyVGE) as a model for lignin, a phenolic copolymer that constitutes the burning fuel in wood (McDermott et al., 1990).

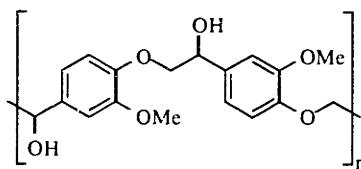


Figure 3.15. Polyveratryl β -Guaiacyl Ether Compound That Serves as a Model for Lignin Reactivity.

The liquefaction of lignin, either thermally or catalytically, is of considerable interest in the recovery of fuels with high oxygen content, as well as precursors to specialty and fine chemicals. Given the data from model compound experiments, the goal is to simulate the depolymerization process to better understand the tradeoffs between raw material structure

and liquefaction operating conditions as product controlling factors, in particular the deleterious formation of char and water as by-products. Two of the components required to perform the necessary simulations are an understanding of the intrinsic kinetics and the size dependency of the degrading polymer on rates of transport within a catalyst particle.

With regard to the intrinsic kinetics, the liquefaction of poly-VGE copolymer, although it results in a large distribution of products of various sizes, is assumed to occur by a small number of chemical transformations operating on polymer substructures along the polymer chain. The strategy of McDermott *et al.* (1990) is to characterize the intrinsic kinetics of stable model monomers that represent the different polymer substructures. From the intrinsic kinetics obtained for the model compounds, it is possible to simulate the coupled reaction processes occurring at various places along a polymer using techniques described below. They measured the pre-exponential factors and activation energies for twelve different reactions—six in a thermal environment and six over a catalyst—involving ten model monomer compounds. Reaction data are summarized in Table 3.2 and model compounds given in Figure 3.16.

Table 3.2. *Intrinsic Chemistry Studied by McDermott et al. (1990).*

<i>Thermal Reactions</i>	$\log A/\text{sec}^{-1}$	$E^*/\text{kcal mol}^{-1}$
VGE \rightarrow acetophenone + guaiacol	5.03	23.9
GGE \rightarrow acetovillone + guaiacol	5.03	20.7
VGE \rightarrow water + vinyl-VGE	1.87	12.5
GGE \rightarrow water + vinyl-GGE	1.87	12.7
vinyl-VGE \rightarrow guaiacol + char	4.03	15.4
vinyl-GGE \rightarrow guaiacol + vinylguaiacol (char)	4.03	15.4
<i>Catalyzed Reactions</i>	$-\log k / \text{sec}^{-1}$	
VGE \rightarrow acetophenone + guaiacol	3.35	
GGE \rightarrow guaiacol + char	2.86	
VGE \rightarrow VGE keto ether	3.11	
GGE \rightarrow GGE keto ether	3.11	
VGE keto ether \rightarrow acetophenone + guaiacol	2.97	
GGE keto ether \rightarrow acetovillone + guaiacol	2.46	

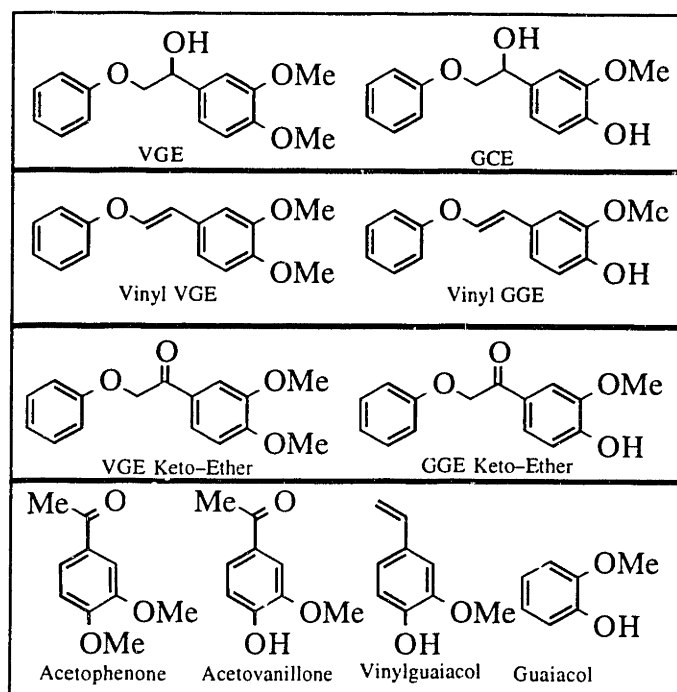


Figure 3.16. Model Compound Used to Obtain Intrinsic Kinetic Parameters In the Liquefaction of Poly-VGE.

As the initial poly-VGE polymer degrades into smaller polymers, the number of compounds present in the reacting system grows exponentially. Thus, a key challenge is how to pose the reaction simulation in a way that both faithfully represents the underlying reaction mechanism and tracks the large number of intermediate species that are formed. The use of unlumped mass action kinetics is prohibitively difficult due to the thousands of intermediate species that are involved. One possible approach is the use of lumped mass action kinetics, which provides a limited amount of information about the product distributions. Even with lumped kinetics, however, there is the problem of parameterizing the equations. Such parameterization would not necessarily allow the use of intrinsic model compound information due to the effects of transport limitations—*i.e.*, lumped kinetics creates a separate parameter estimation problem.

Avoiding these approaches, McDermott *et al.* (1990) exploit the fact that while the number of intermediate species is large, the number of possible transformations are relatively small. Therefore, they develop an approach where the simulation occurs not in the space of chemical concentrations but rather in the much smaller space of chemical transformations and allows the product distribution to be determined as a result.

To accomplish such a simulation, the technique of stochastic kinetics is exploited. In stochastic kinetics, the evolution of the entire depolymerization process is treated as a Markovian process—*i.e.*, a Markov chain that can be generated by Monte Carlo

simulation. The individual transformations that collectively make up the Markov chain are treated as binary random fields whose transition probabilities may be determined from the pre-exponential factors and the rate constant. For example, in a case $A \rightarrow B$ such as the interconversion of enol forms in a segment of a poly-VGE molecule, the transition probability is Eq. 14

$$P_{AB} = 1 - \exp\{-k_{AB}\Delta t\}, \quad (14)$$

where Δt is a time increment of the simulation and k_{AB} is the rate constant determined from the appropriate model compound data. In the interconversion of enols, the rate constant would correspond to VGE \rightarrow VGE keto ether.

The simulation itself consists of a series of time steps, each of length Δt , at which all of the available reaction sites are identified and react with their appropriate transition probability. For a particular site which can undergo a conversion $A \rightarrow B$, this means that:

- a random number, rn , is drawn from the unit interval with a random number generator;
- if $rn \leq P_{AB}$, then the reaction is assumed to occur and the chemical transformation takes place, effectively breaking the molecule down into a set of product molecules; and
- if $rn > P_{AB}$, the reaction does not occur and no change is made.

In addition to allowing the use of intrinsic kinetic information, stochastic kinetics also allows an accounting of transport limitations as a function of the size of intermediate products. For example, in the simulation of lignin liquefaction, McDermott *et al.* (1990) incorporated an effectiveness factor correction to the intrinsic rate constants to account for diffusional limitations in the presence of a catalyst. The effectiveness factor was determined from the Thiele modulus as a function of the molecular weight of the reaction species. As a result, it was possible to study the degree of polymerization, monomer yields, and the molar yield of water as a function of the Thiele modulus of a catalyst.

The method of McDermott *et al.* (1990) shows that chemical modeling can be used to not only to evaluate a reaction path, but also to provide information that is useful in the preliminary design of a chemical process or engineering system. When the modeling and database tools are available, computer calculations are an inherently economical and rapid way of obtaining needed design information.

3.5.2. Systematic Use of Chemical Databases: Industrial Case Study

The last example considered in this chapter is an industrial case study involving the use of a chemical database to develop an environmentally improved reaction path. With over a million chemistry-related journal articles published each year on an extremely diverse set of compounds and reaction classes, perhaps no field of science or engineering spawns more raw information than chemistry. A basic limitation of the approaches to retrosynthetic analysis discussed above is that they require the alternative chemical conversions to be effectively specified in advance. Furthermore, in the approaches due to Powers *et al.* (1975) and Gasteiger *et al.* (1990), quantitative models are not available for prediction of most reaction rates. Many reactions appear to be too sensitive to the nature of the substrate and the reaction condition to be amenable to a simple statistical treatment. For these reasons, an additional information source that has proven to be very useful in process development is the chemical database. A chemical database provides either citations to the chemical literature or direct information with citations for a variety of chemical properties, compound classes, reaction types, and specific transformations. One can often gain insights into possible new chemical reactions by analogy to similar reactions that have been performed in the literature. Additionally, numerous databases such as the Chemical Properties Databank are dedicated to compound properties for risk assessment, modeling, and simulation (Kaufman and Drago, 1993). Such databases are now widely available through on-line services such as DIALOGUE and STN (Kaufman and Drago, 1993). An additional source of such information are CD-ROM packages that incorporate new chemical literature, usually on a quarterly basis (Springer-Verlag, 1992). One of the most useful general purpose databases is the Chemical Abstract Service (CAS) database, which is a computerized version of Chemical Abstracts' compilation of the literature for 1967–present (Fisanick *et al.*, 1994). Previous years are continually being added to a database termed CAOLD and should eventually be available back to the first issues of Chemical Abstracts.

A reaction path synthesis strategy for a class of substrate sensitive formylation reactions (*i.e.*, synthesis of aldehydes) has been undertaken (by the author) in an industrial project with Polaroid Corporation. Due to the substrate sensitive nature of the problem, a CAS database search plays an important role in searches combining the compound class of interest with particular reaction types. The basic elements of the strategy are illustrated in Figure 3.17.

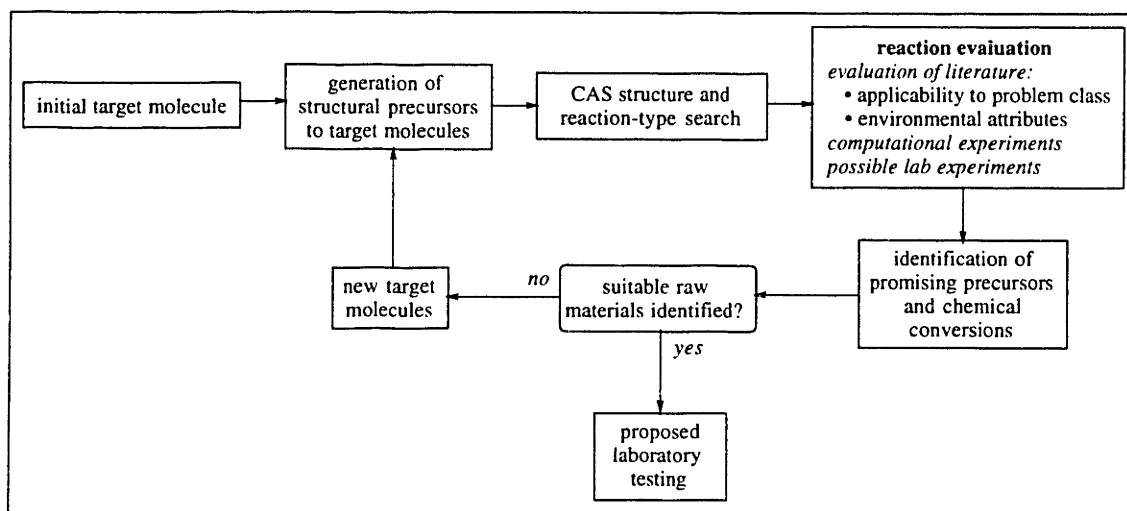
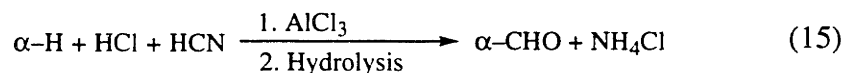


Figure 3.17. Depiction of an Interactive Retrosynthetic Strategy Which Combines CAS Database Searches, Detailed Evaluation of the Literature Sources, and Independent Evaluations.

The motivating case study for the strategy was development of an environmentally improved reaction path for an out-sourced drug intermediate. The traditional route consisted of the Gatterman synthesis as depicted in Eq. 15.



In addition to the safety hazard and cleanup requirements associated with HCN, the reaction involves high pressure vessels as well as considerable additional waste cleanup problems. The main goals of the project were to identify one or more candidate routes that

- minimize safety hazards;
- avoid chlorinated solvents;
- avoid Lewis acid catalysts such as AlCl_3 ; and
- require relatively inexpensive raw materials.

Several results of the computational analysis are illustrated in Figure 3.18.

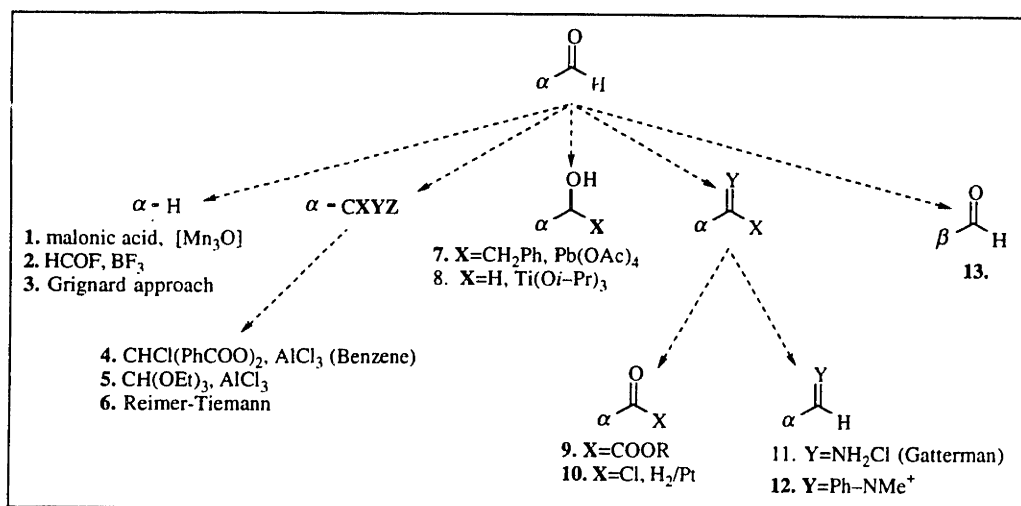
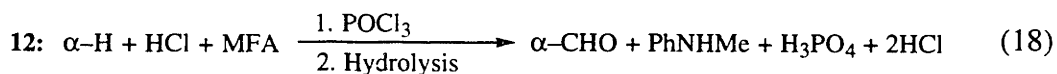
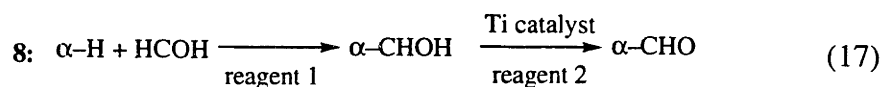
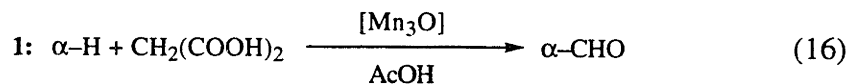


Figure 3.18. Results of the CAS Search for Candidate Formylations of a Targeted Drug Intermediate Substrate, α .

A set of references to the chemical conversions is given in appendix two. Routes 1–3 represent direct formulations of the substrate molecule by a free-radical mechanism, addition of the aldehyde by a Lewis acid catalyst, and by use of a Grignard reagent, respectively. Routes 4–6 involve condensations of different substrates by a precursor of the form —CXYZ. Route 7 is a free radical mechanism involving the cleavage of X by lead tetraacetate. Route 8 involves oxidation of a primary alcohol using a non-toxic titanium alcoholate catalyst. Route 9 is a condensation. Route 10 is the hydrogenation of an acyl halide. Route 11 is the Gatterman synthesis which is the current industrial route (note: the Gatterman-Koch synthesis is not applicable to this substrate). Route 12 is the Vilsmeier-Haack reaction. Of the 13 routes, three could potentially satisfy the environmental and economic requirements discussed above as well as proceed in relatively high yields. These three are given in Eqs. 16–18.



The principle advantage of the first route is the nontoxic solvent. The manganese acetate catalyst, however, is toxic and requires thorough recovery. The disadvantages of

the route are the cost of malonic acid raw material and the toxicity of the manganese catalyst. In order to investigate the reaction feasibility, an empirical ionization potential criteria from the literature was used. Nishino *et al.* (1989) have reported that for the class of compounds under consideration, the substrate ionization potential must be below 7.4 eV for the reaction (Eq. 18) to occur. Since the ionization potential of the substrate under consideration was not reported in their paper, *ab initio* molecular orbital calculations have been utilized. Comparison of computational predictions against measured ionization potentials for related compounds showed that Koopman's theorem did not give accurate values. However, near quantitative accuracy (within 0.25 eV) can be achieved at the Hartree-Fock 3-21G* level using a second order Møller-Plesset correction for the correlation energy. The predicted ionization potential is approximately 7.8 eV. Thus it has been concluded that the substrate is unlikely to undergo formylation as desired, but is more likely to be oxidized to the carboxylic acid.

The Vilsmeier-Haack reaction was an early contender on the basis of suitable existing equipment at Polaroid. An intrinsic limitation of the method is the high cost of methyl formamide (MFA), although recovery and by-product utilization strategies were identified. Further literature review, however, reveals that the reaction has twice been tried previously on the substrate of interest, but in both cases has resulted in low yield.

The condensation reaction has judged the most promising for several reasons:

- the oxidation step is known to proceed to >90% yield in the laboratory (Krohn *et al.*, 1990);
- the formaldehyde raw material is inexpensive;
- the catalyst for the process is non-toxic and relatively inexpensive; and
- non-chlorinated solvents have been employed as reaction media.

A key issue to be addressed in future experimentation is the need for zeolite absorbent to eliminate generated water from the reacting system.

3.6. Discussion

The review of the industrial reaction path synthesis techniques illustrates considerable success in the application of ideas derived from traditional systems engineering algorithms. In general, the limitation of these methods has not been in the "topological" representation of design alternatives but rather the information requirements needed to evaluate them. The approach of Gasteiger *et al.* (1990) has come the closest to an integrated approach to the acquisition of chemical information and evaluation of reaction paths. A basic limitation, however, seems to be a reliance on generalized techniques for the evaluation of new reactions through bond polarity and other types of arguments. While the approach may lead to new insights for the discovery chemist, it seems incongruent with simultaneous

estimation of thermochemical properties for process development. It could be argued that there is a need for a clear distinction between chemical conversions developed in the laboratory and the estimation of process- or environment-related properties. Among the most promising types of approaches are chemical database-driven strategies for retrosynthetic analysis. In addition to the example presented in this chapter, commercial packages are available that perform the retrosynthetic analysis explicitly (*e.g.*, REACCS; Mills *et al.*, 1988).

The example of model compound experimentation introduces a key theme of the thesis that is expanded upon in the next chapter. Namely, in an industrial framework the choice in modeling involves a hierarchy of decisions including

- the choice of empirical versus *ab initio* techniques;
- the choice of model compound experiments (less expensive) versus direct system measurements (more expensive and the most accurate); and
- optimization of the number and generality of adjustable parameters.

A key message is that the incorporation of fundamental *ab initio*-derived parameters in modeling does not eliminate laboratory experiments, but does change the set of experiments to model building experiments required to assimilate the fundamental parameters. It is expected that in the foreseeable future there will still be the need to test computational results to insure that they are accurate and relevant to the problem at hand. The following three chapters propose a modeling approach designed to address the need for integration in the computation of fundamental parameters, construction of empirical models, and use of model compound experiments.

Chapter 4

Quantum Chemical Calculations for the Estimation of Fundamental Molecular Parameters in Model Development

4.1. Introduction

The development of chemical property models is essential to the environmental and economic evaluation of chemical reaction paths. These models are needed in such diverse tasks as predicting reaction feasibility of a substrate, screening partition coefficients of replacement solvents, finding suitable mass separation agents, and predicting synergistic effects (*e.g.*, azeotropes) in complex mixtures. To understand the setting in which these models are used, it is useful to keep in mind that the selection among alternate chemical processes is inherently open-ended. The path leading to a chemical process is an evolutionary one, involving quite a few intermediate decisions as well as the allocation of resources over an extended period of time. No single evaluation criterion, and hence no single model or measurement, is usually decisive for screening alternate reaction paths. A realistic evaluation of a chemical reaction path involves multiple aspects of the physical and chemical properties of the reactants and reaction products, as well as the intrinsic feasibility of the reaction. While the literature is replete with models and suggested experimental measurements for different properties, relatively little attention has been given to the integration the diverse model building and measurement activities that occur in process development. Two key factors are that (*i*) laboratory resources are typically at a premium; and (*ii*) information and design constraints must be communicated in appropriate detail between chemists and engineers. As a result, key goals of a model-building strategy are to

- streamline information sources;
- develop models capable of both aggregation (approximation) and refinement;
- tailor models to specific compound classes so as to minimize model complexity and maximize accuracy; and

- exploit a theoretical basis where possible to maximize transferability of model parameters.

The last goal is of particular importance since physical properties governing a chemical process are often tightly coupled to molecular properties governing the intrinsic chemistry. The measurement or prediction of intrinsic parameters applicable to both thermophysical and molecular phenomena could significantly reduce experimental time and costs. A good example is solvent selection, particularly of replacement solvents. Prediction of solvent effects on reaction rates and impurity reactions could help identify replacement solvents that meet rate and selectivity criteria. At the same time, however, prediction of thermophysical properties is needed to ensure that new solvents do not introduce unacceptable separations problems. Both types of properties can (in some cases) be related to the intrinsic interaction energies between the solvent (*via* pure component or mixture thermodynamic models) and the reaction product constituents (*via* free-energy relationships). Thus, a major challenge in achieving the last goal is to identify computational tools for characterizing common intrinsic parameters.

A tool of growing importance for obtaining fundamental molecular parameters is quantum chemistry. A quantum chemical calculation involves approximating the wave function, $\Psi(\mathbf{r})$, as a function of a local coordinate, \mathbf{r} , for a configuration of atomic nuclei and electrons. Having determined the wave function, a number of useful properties can in principle be estimated, including the minimum energy geometry, electron density ($\Psi^2(\mathbf{r})$), electronic energy levels ($\{\varepsilon_i\}_{i=1}^{n_e}$), electronic frequencies, the dipole moment ($\int d\mathbf{r} \Psi^*(\mathbf{r}) \mathbf{r} \Psi(\mathbf{r})$), and ionization potentials. Quantum calculations have now been applied in a number of engineering systems. One of the landmark industrial applications has been duPont's development of replacement refrigerants (Dixon (1987) provides a general discussion of efforts at duPont). Initial predictions of important molecular properties (*e.g.*, dipole moment, heat capacity) guided the selection of candidate refrigerants to synthesize in the laboratory, ultimately yielding a suitable class of fluorocarbons. Later, when concern was expressed about possible ozone depletion pathways, transition-state calculations were performed to estimate the activation energies of the pathways. Computational results were later backed-up with measured rate constants, which indicated that the pathways were not significant.

The duPont example represents the great advances that have been achieved in the past ten to fifteen years. Only in recent years, with dramatic increases in computational power and improved software, have quantum chemical calculations started to achieve sufficient accuracy and accessibility for use not only in research calculations, but also in industrial research and development. The practical application of quantum calculations, along with molecular-scale simulation techniques such as molecular dynamics and simulated annealing, has now become an intense area of interest both in academic and

industrial research. The ability to compute molecular properties makes it practical to consider predictive models that incorporate these fundamental properties whereas, in the past, obtaining the fundamental properties themselves has been a limiting factor in cost and time. Cited examples of such models using quantum-derived parameters include linear-free energy relationships (McKelvey *et al.*, 1976), pure component thermophysical models (Grigoras, 1991; Murray *et al.*, 1993), partition coefficient models (Grigoras, 1985; Murray *et al.*, 1993), and continuum dielectric models for prediction of solvent effects (Klamt and Schüürmann, 1993). Furthermore, whereas the set of chemical compounds considered has traditionally been constrained by laboratory time and cost, the new computational capabilities allow better screening of a larger number of molecules before going to the laboratory, as illustrated by the duPont example.

This chapter is the first in a series of three chapters that considers a model building strategy incorporating quantum calculations. A brief introduction to quantum chemical calculations is provided based on the Hartree-Fock approximation, which is essential to the study of more advanced computational techniques. A description of properties that can be obtained from the calculations is provided, with an emphasis on those properties that have been used in the model building strategy. Finally, a comparison of the accuracy of different computational methods is given.

4.2. Overview

An introduction to quantum chemistry methods is given with an emphasis on motivating factors in industrial research and development. Numerous practical industrial applications of quantum chemistry fall into two categories:

- gaining conceptual insight into systems where experimental measurements are indecisive; and
- model development in which molecular structure information is synthesized with measured data.

The main thrust of this chapter is to introduce a quantum chemistry background sufficient to understand chapters five and six, wherein a model building strategy that incorporates molecular calculations is developed and verified. In this chapter, a derivation of the Hartree-Fock variational approach to the solution of the time-independent nonrelativistic Schrödinger wave equation is given. A brief discussion of the MNDO and AM1 semi-empirical methods is given and their computational efficiency and robustness discussed. Finally, a discussion of modeling applications is given with an emphasis on the estimation of non-bonding intermolecular interactions.

4.3. Relevant Aspects of Quantum Chemistry

4.3.1. Schrödinger Wave Equation

The quantum theory discussed in this chapter is for non-relativistic time-independent systems. In particular, given an isolated molecule consisting of a set of electrons and fixed atomic nuclei, one wants to predict the ground state electron distribution and energy levels. By extension, the minimum energy geometry of the molecule can be computed through unconstrained optimization techniques. The governing equation is the non-relativistic time-independent Schrödinger wave equation under the Born-Oppenheimer approximation (defined below). A detailed presentation of quantum field theory, which yields the equation as a special case, is beyond the scope of this chapter. A more concise interpretation may be had, however, by a simplified version of the original derivation due to Schrödinger (Weinstock, 1974).

For those familiar with classical dynamics, it is summarized that Schrödinger's derivation consists of a transformation of the Hamilton-Jacobi equation written for a system of n_e electrons and n_A atomic nuclei, the residual of which is then minimized indirectly through a variational formulation (Weinstock, 1974). The solution to the variational problem is a form of the Euler-Lagrange equation termed the Schrödinger wave equation. A derivation that does not require a canonical formalism is presented.

Consider an isolated system of atomic nuclei and electrons in which all interactions between particles are conservative—*i.e.*, the interactions depend only on the current state of the system. The current state of the system is defined in terms of the particle masses, m_i , generalized coordinate vectors, \mathbf{r}_i , and momenta, \mathbf{p}_i , $i = 1, \dots, (n_e + n_A)$. A basic property of an isolated, conservative system is that the sum of the kinetic energy (T) and the potential energy (V) is constant as the system evolves (Eq. 1).

$$T(\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{(n_e+n_A)}) + V(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_{(n_e+n_A)}) = E \quad (1)$$

where

$$T(\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{(n_e+n_A)}) = \sum_{i=1}^{(n_e+n_A)} \frac{\mathbf{p}_i^2}{2m_i} \quad (2)$$

The term E is the total energy of the system. Equations 1 and 2 assume that the kinetic energy is a function of the particle momenta and the potential energy of the particle coordinates, consistent with the conservative nature of the system.

What Schrödinger basically did (Weinstock, 1974) was to introduce a variable transformation for the generalized momenta, the new variable being termed the wave function, Ψ (Eq. 3).

$$p_i = \hbar \frac{\partial \ln \Psi}{\partial r_i}, \quad (3)$$

with the boundary condition

$$\Psi \rightarrow 0 \text{ as } |\mathbf{r}_i| \rightarrow \infty, \quad i = 1, \dots, (n_e + n_A), \quad (4)$$

where $\hbar \equiv h/2\pi$ (h is the Planck constant; $h \equiv 6.616 \dots E-34 \text{ J s}$). In general, the wave function is complex-valued and represents both the mass and wave character of the subatomic particles (Levine, 1991). The most intuitive interpretation of the wave function is as giving rise to the probability of finding a particle in a region of space (recall that the Heisenberg uncertainty principle states that it is impossible to measure both the position and momentum of a particle simultaneously). In particular, the probability of finding a set of particles in jointly configured in respective regions of space D is given by Eq. 5.

$$\int_D d\mathbf{r}_i \int_{R^*} \Psi^* \Psi \prod_{i \neq j}^{(n_e + n_A - 1)} d\mathbf{r}_i, \quad R^* = R^{3(n_e + n_A - 1)}, \quad (5)$$

where the asterisk denotes the complex conjugate. Consequently, the integral over all space is self-normalized:

$$\int \Psi^* \Psi \prod_{i=1}^{(n_e + n_A)} d\mathbf{r}_i = 1. \quad (6)$$

The transformation is mathematically useful, for among other reasons, because it allows the Eq. 1 to be formulated as an eigenvalue problem, with eigenfunctions denoted by Ψ and eigenvalues by E . The spectrum of eigenfunctions and eigenvalues obtained represent the quantized energy states of the system.

Substituting Eq. 3 into Eq. 1 and performing a small number of rearrangements yields Eq. 7.

$$\sum_{i=1}^{(n_e + n_A)} \frac{\hbar^2}{2m_i} \left(\frac{\partial \Psi}{\partial r_i} \right)^2 + (V - E) \cdot \Psi^2 = 0 \quad (7)$$

Rather than solve Eq. 5 directly, the approach laid out by Schrödinger is to formally define the residual for the left hand side of Eq. 7 and then minimize the integral-averaged residual over an arbitrary domain D . The integral is defined as

$$I^* = \int_D f \prod_{i=1}^{(n_e+n_A)} dr_i, \quad (8)$$

where the residual f is defined in Eq. 9.

$$f \equiv \sum_{i=1}^{(n_e+n_A)} \frac{\hbar^2}{2m_i} \left(\frac{\partial \Psi}{\partial r_i} \right)^2 + (V - E) \cdot \Psi^2 \quad (9)$$

Appealing to the arbitrariness of the domain of integration, it can be shown that extremization of Eq. 5 insures that Eq. 4 is satisfied to within some measure-theoretic qualifications. Neglecting the notation that goes along with those qualifications, it is shown in effect that Eq. 10 holds.

$$\frac{\partial I^*}{\partial r_i} = 0, \quad i = 1, \dots, (n_e + n_A) \Rightarrow f \equiv 0 \quad (10)$$

Performing the differentiations analytically yields a differential equation known as the Euler-Lagrange equation:

$$\frac{\partial f}{\partial \Psi} - \sum_{i=1}^{(n_e+n_A)} \frac{\partial}{\partial r_i} \cdot \frac{\partial f}{\partial \Psi_i} = 0 \quad (11)$$

where

$$\Psi_i \equiv \frac{\partial \Psi}{\partial r_i}. \quad (12)$$

Substituting Eq. 7 into the Euler-Lagrange equation and separating the particles into the set of electrons and atomic nuclei yields the Schrödinger wave equation, which is given by Eq. 13.

$$\sum_{i=1}^{n_e} \frac{\hbar^2}{2m_e} \nabla_i^2 \Psi + \sum_{A=1}^{n_A} \frac{\hbar^2}{2m_A} \nabla_A^2 \Psi + (E - V) \Psi = 0, \quad (13)$$

where

$$\nabla_i^2 \equiv \frac{\partial}{\partial r_i} \cdot \frac{\partial}{\partial r_i}.$$

The term m_e denotes the mass of an electron, m_A the atomic nuclear mass ($A = 1, \dots, n_A$).

Equation 13 is usually rearranged to Eq. 14, revealing that the Schrödinger wave equation is an eigenvalue problem for the Hamiltonian operator, H (Eq. 15).

$$H \Psi = E \Psi \quad (14)$$

where

$$H \equiv - \sum_{i=1}^{n_e} \frac{\hbar^2}{2m_e} \nabla_i^2 - \sum_{A=1}^{n_A} \frac{\hbar^2}{2m_A} \nabla_A^2 + V \quad (15)$$

The Hamiltonian operates on the wave function to generate the kinetic energy contributions (first two terms) and the potential energy contributions. The potential energy contributions are due to coulombic interactions between the particles. In the absence of external fields, three types of interactions are present:

- electron-nucleus attractions (Eq. 16; given in atomic coordinates; for a table of applicable unit conversions, see Szabo and Ostlund, 1989);

$$V_{e-N} \equiv \sum_{i=1}^{n_e} \sum_{A=1}^{n_A} \frac{Z_A}{r_{iA}} \quad (16)$$

r_{iA} —distance between electron i and nucleus A

Z_A —net charge of nucleus A

- electron-electron repulsions (Eq. 17);

$$V_{e-e} \equiv \sum_{i=1}^{n_e} \sum_{j>i}^{n_e} \frac{1}{r_{ij}} \quad (17)$$

r_{ij} —distance between electron i and electron j

- and nuclear-nuclear repulsions (Eq. 18).

$$V_{N-N} \equiv \sum_{A=1}^{n_A} \sum_{B>A}^{n_A} \frac{Z_A Z_B}{r_{AB}} \quad (18)$$

r_{AB} —distance between nucleus A and nucleus B

The Schrödinger wave equation can be simplified considerably by exploiting the fact that the atomic nuclei are much more massive than the electrons (the ratio of electron mass to proton mass is about $5.45E-4$) and move much more slowly. Consequently, the Born–Oppenheimer approximation states that the electrons can be treated as moving in a constant field generated by fixed nuclei. For the special case of fixed nuclei, the problem of determining molecular electronic structure becomes the eigenvalue problem given in Eq. 19.

$$H_e \Psi_e = E_e \Psi_e \quad (19)$$

where

$$H_e \equiv - \sum_{i=1}^{n_e} \frac{1}{2} \nabla_i^2 - \sum_{i=1}^{n_e} \sum_{A=1}^{n_A} \frac{Z_A}{r_{iA}} + \sum_{i=1}^{n_e} \sum_{j>i}^{n_e} \frac{1}{r_{ij}}, \quad (20)$$

$$E = E_e + \sum_{A=1}^{n_A} \sum_{B>A}^{n_A} \frac{Z_A Z_B}{r_{AB}}, \quad (21)$$

and

$$\int \Psi_e^2 = n_e \quad (22)$$

Thus, the molecular electronic structure problem depends explicitly on the coordinates of the electrons and parametrically on the coordinates of the nuclei. Equation 19 is the starting point for the computational quantum chemistry of many-electron molecules and in particular the Hartree-Fock approximation. Since the concepts introduced in the Hartree-Fock approximation transfer to most computational methods currently available (an exception is density functional theory; Levine, 1991), a detailed derivation of the approximation is considered in the next subsection.

4.3.2. Solution Structure

There are an infinite number of solutions to Eq. 19. Denoting the eigenfunctions by

$$\{\chi_k\}_{k=1}^{\infty} = \{\chi_1, \chi_2, \dots, \chi_{n_e}, \chi_{n_e+1}, \dots\}$$

and the corresponding eigenvalues by

$$\{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_{n_e}, \varepsilon_{n_e+1}, \dots \mid \varepsilon_1 < \varepsilon_2 < \dots\},$$

the general solution of the ground state (*i.e.*, lowest energy) wavefunction, Ψ_{gs} , can be shown to have the form given by Eq. 23.

$$\Psi_{gs} = \sum_{j=1}^{\infty} c_j \Psi_j \left[\{\chi_k\}_{k \in K_j} \right], \text{ card}(K_j) = n_e \quad (23)$$

Correct specifications of the eigenfunctions $\{\chi_k\}_{k=1}^{\infty}$ and the functionals $\{\Psi_j\}_{j=1}^{\infty}$ (termed "configuration state functions") involve an important constraint in addition to the Schrödinger wave equation. Specifically, electrons, along with other elementary particles, are endowed with an intrinsic angular momentum above and beyond that due to the "orbit" around the atomic nucleus. This momentum is termed the spin angular momentum (Levine, 1991), but is not directly analogous to any classical effect. The spin states of electrons in a molecule affect observable spectral and magnetic properties of molecules. Thus, the spin dependency clearly must be incorporated into the wave function. It can be shown (Levine, 1991) that the eigenfunctions must satisfy the eigenvalue problem defined by Eq. 24.

$$S_z \chi_k = m_s^k \hbar \chi_k, \quad s = 1/2, \quad m_{1/2}^k = \pm 1/2, \quad k = 1, 2, \dots, \infty, \quad (24)$$

where s is the intrinsic half-integral spin of an electron, m_s defines the "spin up" and "spin down" quantum states, and S_z is the operator for the "z - component" of the spin angular momentum. Clearly there are two solutions to the problem, corresponding to $m_{1/2} = -1/2$ and $1/2$.

Analogously, the wavefunction must satisfy Eq. 25.

$$S_z \Psi_{gs} = \frac{1}{2} (N^\alpha - N^\beta) \Psi_{gs}, \quad (25)$$

where N^α is the number of electrons in the "spin up" state and N^β the number in the "spin down" state. The case where $N^\alpha - N^\beta = 0$ corresponds to a spin-paired molecule, termed a singlet, whereas $|N^\alpha - N^\beta| \neq 0$ indicates a radical, usually either a doublet or a

triplet. A key connection between Eqs. 24 and 25 is that all approximations to Ψ_{gs} in terms of the spin eigenfunctions should satisfy Eq. 25. Furthermore, since the spin operator does not involve spatial coordinates and the Schrödinger equation does not involve m_s (for the nonrelativistic case), it follows that the Hamiltonian and spin operators should commute, whence the eigenfunctions can be separated into spatial and spin contributions. The appropriate eigenfunction structure is given in Eq. 26.

$$\chi_k(\mathbf{x}) = \alpha(\omega) \psi_k(\mathbf{r}) \text{ or } \beta(\omega) \psi_k(\mathbf{r}), \mathbf{x} = \langle \mathbf{r}, \omega \rangle, k = 1, \dots, n_e, \quad (26)$$

where $\psi_k(\mathbf{r})$ is the "spatial orbital" of the Cartesian vector, \mathbf{r} , and ω is a dummy variable associated with the spin eigenfunctions α and β (for $m_{1/2} = -1/2$ and $1/2$, respectively). The eigenfunction χ_k is therefore termed a "spin orbital."

For molecules in the singlet state, it is usually assumed that all electrons are spin paired—*i.e.*, there are $n_e/2$ spatial orbitals, each containing two electrons of opposite spin. The spin orbitals are constructed as follows:

$$\chi_{2k}(\mathbf{x}) = \alpha(\omega) \psi_k(\mathbf{r}), k = 1, \dots, n_e/2 \quad (27)$$

$$\chi_{2k-1}(\mathbf{x}) = \beta(\omega) \psi_k(\mathbf{r}), k = 1, \dots, n_e/2. \quad (28)$$

Equations 27 and 28 are very good approximations in most cases. Exceptions include unusually long bond lengths such as those encountered in the calculation of potential energy surfaces for dissociation reactions. In such cases, as well as for doublet and triplet states, a unique spatial orbital is often assigned to each electron.

The spin angular momentum also constrains the structure of the configuration state functions. Let the form of the functionals be denoted by Eq. 29.

$$\Psi_j = \left| \chi_{k_1}(\mathbf{x}_1) \chi_{k_2}(\mathbf{x}_2) \cdots \chi_{k_{n_e}}(\mathbf{x}_{n_e}) \right\rangle, k_l \in K_j, \quad (29)$$

where the $1, 2, \dots, n_e$ electrons are randomly assigned to the spin orbitals (the full implication of the bracket notation will not be pursued here; Szabo and Ostlund (1989) provide an introduction). Due to the Heisenberg uncertainty principle, it is not possible to determine the precise evolution of the electrons in a molecule. More specifically, it is not possible to uniquely assign an electron to the spatial orbital χ_k . For example, suppose the first and second electrons are interchanged, or permuted (Eq. 30).

$$P_{k_1 k_2} \left| \chi_{k_1}(\mathbf{x}_1) \chi_{k_2}(\mathbf{x}_2) \cdots \chi_{k_{n_e}}(\mathbf{x}_{n_e}) \right\rangle = \left| \chi_{k_1}(\mathbf{x}_2) \chi_{k_2}(\mathbf{x}_1) \cdots \chi_{k_{n_e}}(\mathbf{x}_{n_e}) \right\rangle \quad (30)$$

where $P_{k_1 k_2}$ is the operator that permutes electrons in orbitals k_1 and k_2 . If the permutation is performed again, the resulting configuration state function must be identical to Eq. 29, *i.e.*, Eq. 31 must hold.

$$P_{k_1 k_2} P_{k_1 k_2} \left| \chi_{k_1}(\mathbf{x}_1) \chi_{k_2}(\mathbf{x}_2) \cdots \chi_{k_{n_e}}(\mathbf{x}_{n_e}) \right\rangle = \Psi_j \quad (31)$$

There are only two ways that Eq. 31 can be satisfied: either the permutation of two electrons is symmetric, *i.e.*, Eq. 32 must hold,

$$\left| \chi_{k_1}(\mathbf{x}_1) \chi_{k_2}(\mathbf{x}_2) \cdots \chi_{k_{n_e}}(\mathbf{x}_{n_e}) \right\rangle = \left| \chi_{k_1}(\mathbf{x}_2) \chi_{k_2}(\mathbf{x}_1) \cdots \chi_{k_{n_e}}(\mathbf{x}_{n_e}) \right\rangle, \quad (32)$$

or it is antisymmetric, *i.e.*, Eq. 33 must hold,

$$\left| \chi_{k_1}(\mathbf{x}_1) \chi_{k_2}(\mathbf{x}_2) \cdots \chi_{k_{n_e}}(\mathbf{x}_{n_e}) \right\rangle = - \left| \chi_{k_1}(\mathbf{x}_2) \chi_{k_2}(\mathbf{x}_1) \cdots \chi_{k_{n_e}}(\mathbf{x}_{n_e}) \right\rangle. \quad (33)$$

According to quantum theory, the symmetry or antisymmetry of the wave function for a particle is governed by the spin states available to the particle. Particles with half-integral spin states such as electrons are known as fermions (Levine, 1991). It turns out that fermions are required to have antisymmetric wave functions. Thus, the following principle must be adhered to in the construction of the configuration state functions:

Pauli Exclusion Principle: The many-electron wavefunction must be antisymmetric with respect to the interchange of any two electrons.

Since the solution to the wave equation is an eigenvalue problem, one expects the functionals to involve appropriately weighted products of the spin orbitals. The simple product of spin eigenfunctions, however, is a symmetric function and therefore does not satisfy the Pauli exclusion principle. A form that does satisfy the Pauli exclusion principle is the determinant operator. Determinants can be constructed from the spin orbitals as illustrated for $n_e = 3$ in Eq. 34.

$$\left| \chi_{k_1}(\mathbf{x}_1) \chi_{k_2}(\mathbf{x}_2) \chi_{k_3}(\mathbf{x}_3) \right\rangle \equiv (n_e!)^{-1/2} \begin{vmatrix} \chi_{k_1}(\mathbf{x}_1) & \chi_{k_2}(\mathbf{x}_1) & \chi_{k_3}(\mathbf{x}_1) \\ \chi_{k_1}(\mathbf{x}_2) & \chi_{k_2}(\mathbf{x}_2) & \chi_{k_3}(\mathbf{x}_2) \\ \chi_{k_1}(\mathbf{x}_3) & \chi_{k_2}(\mathbf{x}_3) & \chi_{k_3}(\mathbf{x}_3) \end{vmatrix}, \quad k_l \in K_j \quad (34)$$

Equation 34 is referred to as a Slater determinant (Szabo and Ostlund, 1989). The term $(n_e!)^{-1/2}$ is a normalizing factor.

Based on Eq. 34, the functionals can be constructed from the set of spin orbitals to create all unique Slater determinants. As a practical matter, careful choice of the sets K_0, K_1, K_2, \dots is motivated by the need to identify those determinants that contribute the most to the wavefunction. One useful principle is that the occupation of different energy levels by electrons is subject to the Boltzmann distribution, leading to the expectation that the lowest energy orbitals are the ones that make the most significant contribution to the wavefunction. Therefore, the most significant contributor to the ground state energy, Ψ_0 , should be the Slater determinant consisting of spin orbitals corresponding to the n_e smallest (most negative) eigenvalues. A rational way to define the remaining functionals is to replace a specified number of the lowest energy orbitals with higher energy, excited state spin orbitals. For example,

$$\Psi_1 \equiv \sum_{k_j=1}^{n_e} \sum_{k_p=n_e+1}^{\infty} c_{k_j k_p} \Psi_0^{k_j \rightarrow k_p} \quad (35)$$

where $k_j \rightarrow k_p$ indicates that the ground state spin orbital χ_{k_j} is replaced by the orbital χ_{k_p} in the Slater determinant. The parameter $c_{k_j k_p}$ is a weighting coefficient. The determinants $\Psi_0^{k_j \rightarrow k_p}$ are termed "singly excited determinants" because they contain spin orbitals corresponding to excited energy states.

With the development so far, it is possible to begin discussing approximations of Schrödinger wave equation that are suitable for computational solution (solution strategies are discussed in the next section). The starting point for most approaches is the Hartree-Fock approximation, which approximates the ground state wavefunction as $\Psi_{gs} \equiv \Psi_0$, where Ψ_0 is the single Slater determinant consisting of the ground state spin orbitals. The resulting eigenvalue problem is given by Eq. 36.

$$H_e \Psi_0 = E_0 \Psi_0 \quad (36)$$

where the Hartree-Fock energy, E_0 , provides an upper bound on the ground state energy of the system.

The scope of Hartree-Fock approximation can be understood in terms of the treatment of electron-electron interactions. The motions of two electrons are correlated due to two distinct effects. First, the antisymmetry principle induces a correlation of motion between electrons of like spin, resulting in a "Fermi hole" surrounding each electron. A

Fermi hole denotes a reduced joint probability of finding two electrons of like spin in proximity of one another. Second, the electron-electron repulsion potential between two electrons requires that the probability of finding two or more electrons at the same spatial point is zero. The Hartree-Fock wave function incorporates the former but not the latter, as can be demonstrated with the two-electron Slater determinant (Eq. 37).

$$\Psi_0(\mathbf{x}_1, \mathbf{x}_2) = |\chi_1(\mathbf{x}_1)\chi_2(\mathbf{x}_2)\rangle = \frac{1}{\sqrt{2}}[\chi_1(\mathbf{x}_1)\chi_2(\mathbf{x}_2) - \chi_2(\mathbf{x}_1)\chi_1(\mathbf{x}_2)], \quad (37)$$

Recalling that the probability of finding the two electrons in a differential element $d\mathbf{r}_1 d\mathbf{r}_2$ is given by

$$P(\mathbf{r}_1, \mathbf{r}_2) = \left[\int d\omega_1 d\omega_2 |\Psi_0|^2 \right] d\mathbf{r}_1 d\mathbf{r}_2, \quad (38)$$

the correlation due to the antisymmetry principle can be illustrated by assuming that the two electrons have opposite spin (Eq. 39).

$$\chi_1(\mathbf{x}_1) = \psi_1(\mathbf{r}_1)\alpha(\omega_1), \quad \chi_2(\mathbf{x}_2) = \psi_2(\mathbf{r}_2)\beta(\omega_2). \quad (39)$$

Substituting Eqs. 37 and 39 into Eq. 38 yields

$$P(\mathbf{r}_1, \mathbf{r}_2) = \frac{1}{2} \left[|\psi_1(\mathbf{r}_1)|^2 |\psi_2(\mathbf{r}_2)|^2 + |\psi_1(\mathbf{r}_2)|^2 |\psi_2(\mathbf{r}_1)|^2 \right] d\mathbf{r}_1 d\mathbf{r}_2. \quad (40)$$

That the positions of the two electrons are uncorrelated follows from the fact that the joint probability is a sum of separable probabilities (*i.e.*, of the form $P_1(\mathbf{r}_1) \cdot P_2(\mathbf{r}_2)$) operating on interchangeable, indistinguishable electrons. In particular, the predicted probability that two electrons of opposite spin occupy the same location is non-zero.

If, however, the two electrons have identical spins, the resulting probability distribution is given by Eq. 41.

$$P(\mathbf{r}_1, \mathbf{r}_2) = \frac{1}{2} \left[|\psi_1(\mathbf{r}_1)|^2 |\psi_2(\mathbf{r}_2)|^2 + |\psi_1(\mathbf{r}_2)|^2 |\psi_2(\mathbf{r}_1)|^2 - \right. \\ \left. - \left\{ \psi_1^*(\mathbf{r}_1) \psi_2(\mathbf{r}_1) \psi_2^*(\mathbf{r}_2) \psi_1(\mathbf{r}_2) + \psi_1(\mathbf{r}_1) \psi_2^*(\mathbf{r}_1) \psi_2(\mathbf{r}_2) \psi_1^*(\mathbf{r}_2) \right\} \right] d\mathbf{r}_1 d\mathbf{r}_2 \quad (41)$$

The cross-terms in Eq. 41 are not separable and effectively reduce the probability of finding two electrons of identical spin in the same vicinity. In particular, the joint probability is zero for $\mathbf{r}_1 = \mathbf{r}_2$, corresponding to the Fermi hole.

The failure of the Hartree-Fock approximation to account for correlation due to electron-electron repulsion (Eq. 17) is a real limitation of the method. Absolute energies

(but not necessary relative energies) predicted by the Hartree-Fock approximation tend to be significantly in error with respect to spectroscopic measurements. The limited treatment of correlation energy also leads to significant deviations in observable properties, including binding and dissociation energies. Furthermore, the Hartree-Fock method is not suitable for estimation of dispersion-type interactions between molecules, which arise from electron-electron correlation.

Having stated these limitations, the Hartree-Fock method and its underlying formalism remain of central importance in computational quantum chemistry. For example, electron correlation can be estimated through addition of higher configuration state functions as perturbations to the single determinant Hartree-Fock configuration state function. Furthermore, the excited-state spin orbitals required to construct the additional configuration state functions can in fact be obtained from the original Hartree-Fock equation. Other approaches to incorporate the correlation energy, such as the Møller-Plesset perturbation theory (Szabo and Ostlund, 1989), also use the Hartree-Fock configuration state function. If the optimal geometry of a molecule is not known, it can be estimated by adjusting the positions of the atomic nuclei to minimize the Hartree-Fock energy.

4.3.3. Solution of the Hartree-Fock Approximation

The most common, though not the only, strategy for solving Eq. 36 is to minimize the ground state energy, E_0 , through a variational formulation. An explicit integral formula for the ground state energy is obtained by pre-multiplying Eq. 36 by Ψ_0 and integrating, to yield Eq. 38.

$$\int \Psi_0^* H_e \Psi_0 \prod_{i=1}^{n_e} dx_i = E_0 \int \Psi_0^* \Psi_0 \prod_{i=1}^{n_e} dx_i = E_0 \quad (42)$$

Implicit in Eq. 42 is the normalization of the wavefunction, which can be expressed either in terms of the wavefunction,

$$\int \Psi_0^* \Psi_0 \prod_{i=1}^{n_e} dx_i = 1, \quad (43)$$

or the spin orbitals themselves,

$$\int \chi_a^*(x_1) \chi_b(x_1) dx_1 = \delta_{ab}, \quad \delta_{ab} = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases}, \quad a, b = 1, \dots, n_e, \quad (44)$$

where δ_{ab} is the Kronecker delta.

The minimization of E_0 may be accomplished through the method of undetermined multipliers. The objective of the procedure is to minimize Eq. 42 subject to the normalization constraints (Eq. 44). The Lagrangian of the problem is given by Eq. 45.

$$L = E_0 \left[\{\chi_k\}_{k=1}^{n_e} \right] - \sum_{a=1}^{n_e} \sum_{b=1}^{n_e} \varepsilon_{ab} \left(\int \chi_a^*(x_1) \chi_b(x_1) dx_1 - \delta_{ab} \right), \quad (45)$$

where the ε_{ab} are Lagrange multipliers for the normalization constraints. For the ground state energy to be a local minimum, the following stationary conditions must be satisfied simultaneously with Eq. 44.

$$\frac{\partial L}{\partial \chi_k} = 0, \quad k = 1, \dots, n_e \quad (46)$$

It can be shown (Szabo and Ostlund, 1989) that the stationary conditions yield Eqs. 47–51.

$$f(x_1) \chi_a(x_1) = \sum_{b=1}^{n_e} \varepsilon_{ab} \chi_b(x_1), \quad a = 1, \dots, n_e \quad (47)$$

where

$$f(x_1) \equiv h(x_1) + \sum_{b \neq a} J_b(x_1) - K_b(x_1) \quad (48)$$

$$\sum_{a=1}^{n_e} \varepsilon_{aa} = E_0, \quad (49)$$

$$J_b(x_1) = \int d\omega dr_2 |\chi_b(x_2)|^2 r_{12}^{-1}, \text{ and} \quad (50)$$

$$K_b(x_1) \chi_a(x_1) = \left[\int d\omega dr_2 \chi_b^*(x_2) r_{12}^{-1} \chi_a(x_2) \right] \chi_a(x_1). \quad (51)$$

The integral operators $J_b(x_1)$ and $K_b(x_1)$ are termed the coulomb and exchange operators. The coulomb operator represents the averaged local repulsion potential at point r_1 due to an electron in a spin orbital χ_b . The interpretation of the exchange integral is less straightforward (unlike the coulomb operator, the exchange operator is not directly associated with a point-wise potential energy surface) and is related to the exchange of two electrons among two spin orbitals. It can be shown that Eq. 47 is diagonalizable to yield a

unique multiplier for each spin orbital. Thus, Eqs. 47 and 49 can be replaced by Eqs. 52 and 53.

$$f(\mathbf{x}_I) \chi_a(\mathbf{x}_I) = \varepsilon_a \chi_a(\mathbf{x}_I), \quad a = 1, \dots, n_e \quad (52)$$

where

$$\sum_{a=1}^{n_e} \varepsilon_a = E_0, \quad (53)$$

The integrations over the spin domain in Eq. 52 can be carried out analytically and independently of the spatial domain. Recall that in the case of a closed-shell molecule, it can be assumed that the n_e spin orbitals are formed from n_e spin eigenfunctions ($N^\alpha + N^\beta = n_e$) which contain $n_e/2$ spatial orbitals. Consequently, for the case of a closed shell molecule, integration of Eq. 52 over the spin domain yields Eqs. 54–57.

$$\hat{f}(\mathbf{r}_I) \psi_i(\mathbf{r}_I) = \varepsilon_i \psi_i(\mathbf{r}_I), \quad i = 1, \dots, n_e/2, \quad (54)$$

where

$$\hat{f}(\mathbf{r}_I) = h(\mathbf{r}_I) + \sum_{a=1}^{n_e/2} 2J_a(\mathbf{r}_I) - K_a(\mathbf{r}_I) \quad (55)$$

$$J_a(\mathbf{r}_I) = \int d\mathbf{r}_2 |\psi_a(\mathbf{r}_2)|^2 r_{I2}^{-1}, \text{ and} \quad (56)$$

$$K_a(\mathbf{r}_I) \psi_i(\mathbf{r}_I) = \left[\int d\mathbf{r}_2 \psi_a^*(\mathbf{r}_2) r_{I2}^{-1} \psi_i(\mathbf{r}_2) \right] \psi_a(\mathbf{r}_I). \quad (57)$$

Computational solution of the integro-differential equation (Eq. 54) to determine approximate spatial wavefunctions $\{\psi_i(\mathbf{r})\}_{i=1}^{n_e/2}$ is accomplished through construction of the Roothaan-Hall equations (Szabo and Ostlund, 1989; if the electrons are not all spin paired, then the appropriate equations are the Pople-Nesbet equations). Let each spatial orbital be approximated by finite expansion in terms of a common set of $K(\geq n_e/2)$ basis functions, $\{\phi_\mu(\mathbf{r})\}_{\mu=1}^K$, as shown in Eq. 58.

$$\psi_i(\mathbf{r}) = \sum_{\mu=1}^K C_{\mu i} \phi_\mu(\mathbf{r}), \quad i = 1, \dots, n_e/2, \dots, K \quad (58)$$

Inserting Eq. 58 into Eq. 54 yields

$$\hat{f}(\mathbf{r}_I) \sum_{\mu=1}^K C_{\mu i} \phi_{\mu}(\mathbf{r}_I) = \varepsilon_i \sum_{\mu=1}^K C_{\mu i} \phi_{\mu}(\mathbf{r}_I) + R_i(\mathbf{r}_I), \quad i = 1, \dots, n_e/2, \dots, K, \quad (59)$$

where $R_i(\mathbf{r}_I)$ is the residual error due to the approximation of $\psi_i(\mathbf{r})$ by Eq. 58. Note that as a by-product of the expansion one obtains not only the $n_e/2$ occupied spatial orbitals, but also an additional $(K - n_e/2)$ "virtual orbitals," which represent unoccupied excited electronic states. The virtual orbitals are directly useful in constructing additional configuration state functions (*e.g.*, approximations to Eq. 35) in configuration interaction techniques. It should be stated that the convention used in several quantum chemistry texts of stating Eq. 52 as an equality without the residual term (*e.g.*, Szabo and Ostlund, 1989) is not correct, since the basis function expansion (Eq. 51) is only an approximation of the true spatial orbital for finite K . More importantly, a correct derivation and interpretation of the Roothaan-Hall equations requires the $R_i(\mathbf{r}_I)$.

Specifically, when one looks at Eq. 59 what is really being represented? Is it correct to say, for example, that Eq. 59 comprises K equations and K^2 variables (*i.e.*, the $\left\{ C_{\mu i} \right\}_{\mu=1, \dots, K}^{i=1, \dots, K}$) and is thus an underdetermined system? The answer is no. In fact, Eq. 59 represents an infinite number of equations: a different equation is generated for each instance of the position vector \mathbf{r}_I . Therefore, one expects that there is no choice of the finite number of coefficients $\left\{ C_{\mu i} \right\}_{\mu=1, \dots, K}^{i=1, \dots, K}$ that will satisfy Eq. 59 (*i.e.*, $R_i(\mathbf{r}_I) = 0$) for all realizations of \mathbf{r}_I . Rather, one expects $|R_i(\mathbf{r}_I)| > 0$ for an infinite number of \mathbf{r}_I .

The goal of the solution procedure is to minimize the residual of approximation in some sense. Several approaches could be tried. One approach, for example, would be to pick a set $\left\{ \mathbf{r}_I^{\beta} \right\}_{\beta=1, \dots, K}$ of points and require that

$$R_i(\mathbf{r}_I^{\beta}) = 0, \quad \beta = 1, \dots, K, \quad i = 1, \dots, K. \quad (60)$$

Minimization of the residual in this way is known as a collocation strategy and is sometimes used in finite element methods. Although the author has not seen the collocation strategy applied to the Hartree-Fock problem, the strategy provides an interesting contrast to the method that is used, which is to minimize the error in the function space spanned by the basis functions themselves (Eq. 61).

$$\int \phi_v^*(\mathbf{r}_I) R_i(\mathbf{r}_I) d\mathbf{r}_I = 0, \quad v = 1, \dots, K, \quad i = 1, \dots, K. \quad (61)$$

In effect, Eq. 61 requires the residual function, which represents the error of the approximation, to be orthogonal to the function space spanned by the basis functions. In the context of finite element and spectral methods for the solution of differential equations, this approach is known as the Galerkin method. For the specific case of the Hartree-Fock approximation, Eq. 57 generates the well-known Roothaan-Hall equations (Szabo and Ostlund, 1989). Thus, the Roothaan-Hall equations are interpreted as a spectral solution to the Hartree-Fock equations *via* Galerkin orthogonalization of the residual.

Carrying out the integrals in Eq. 61, the resulting Roothaan-Hall equations are given by Eqs. 62–64.

$$\sum_{\mu=1}^K F_{v\mu} C_{\mu i} = \epsilon_i \sum_{\mu=1}^K S_{v\mu} C_{\mu i}, \quad i = 1, \dots, K, \quad (62)$$

where

$$F_{v\mu} = \int d\mathbf{r}_I \phi_v^*(\mathbf{r}_I) f(\mathbf{r}_I) \phi_{\mu}(\mathbf{r}_I), \quad \text{and} \quad (63)$$

$$S_{v\mu} = \int d\mathbf{r}_I \phi_v^*(\mathbf{r}_I) \phi_{\mu}(\mathbf{r}_I). \quad (64)$$

The matrix formed from the $\{F_{v\mu}\}_{v=1, \dots, K}^{\mu=1, \dots, K}$ is termed the "Fock matrix" and from the $\{S_{v\mu}\}_{v=1, \dots, K}^{\mu=1, \dots, K}$ the "overlap matrix." It is noted that the overlap matrix is the $K \times K$ identity matrix if the basis functions are mutually orthonormal. The expansion of the Fock operator in Eq. 62 reveals two types of computations to be performed (Eq. 65).

$$F_{v\mu} = H_{v\mu}^{\text{core}} + G_{v\mu}, \quad (65)$$

where $H_{v\mu}^{\text{core}}$ is an element of the "core-Hamiltonian" matrix, which contains the kinetic energy and electron-nuclear attraction contributions:

$$H_{v\mu}^{\text{core}} = \int d\mathbf{r}_I \phi_v^*(\mathbf{r}_I) \left[-\frac{1}{2} \nabla_{\mathbf{r}_I}^2 \right] \phi_{\mu}(\mathbf{r}_I) + \int d\mathbf{r}_I \phi_v^*(\mathbf{r}_I) \left[-\sum_{A=1}^{n_A} \frac{Z_A}{r_{iA}} \right] \phi_{\mu}(\mathbf{r}_I), \quad (66)$$

and $G_{v\mu}$ constitutes contributions due to various electron-electron interactions (Eqs. 67–69):

$$G_{\nu\mu} = \sum_{\lambda=1}^K \sum_{\sigma=1}^K P_{\lambda\sigma} \left[(\nu\mu, \sigma\lambda) - \frac{1}{2}(\nu\lambda, \sigma\mu) \right] \quad (67)$$

where

$$(\nu\mu, \sigma\lambda) \equiv \int d\mathbf{r}_1 d\mathbf{r}_2 \phi_\nu^*(\mathbf{r}_1) \phi_\mu(\mathbf{r}_1) r_{12}^{-1} \phi_\sigma^*(\mathbf{r}_2) \phi_\lambda(\mathbf{r}_2) \quad (68)$$

$$(\nu\lambda, \sigma\mu) \equiv \int d\mathbf{r}_1 d\mathbf{r}_2 \phi_\nu^*(\mathbf{r}_1) \phi_\lambda(\mathbf{r}_1) r_{12}^{-1} \phi_\sigma^*(\mathbf{r}_2) \phi_\mu(\mathbf{r}_2) \quad (69)$$

The term $P_{\lambda\sigma}$ is an element of the "bond-order" matrix, defined by Eq. 70.

$$P_{\lambda\sigma} \equiv 2 \sum_{a=1}^{n_e/2} C_{\lambda a} C_{\sigma a}^* \quad (70)$$

Notice that only the first $n_e/2$ eigenvectors are involved in the bond-order matrix, which corresponds to the occupied spatial orbitals.

The computational complexity associated with the Hartree-Fock method is primarily due to large number of two-electron integrals (Eqs. 68 and 69) that must be evaluated in Eq. 67. The number of two electron integrals is a permutation of possible quartets of basis functions. Under the assumption that the basis functions are real-valued, the total number to be evaluated goes as $O(K^4/8)$ (e.g., for $K=100$ the total number of integrals is 12,753,775; this basis set would be typical of a medium sized organic molecule consisting of first row atoms, say with a molecular weight of 100). Even with analytic evaluation of the integrals, storage and manipulation of the set of integrals is demanding for large systems.

The Hartree-Fock approximation requires simultaneous solution of the K eigenvalue problems defined by Eq. 62. In practice, these equations are concatenated and solved simultaneously for the eigenvectors $\{\{C_{1i}, C_{2i}, \dots, C_{Ki}\}\}_{i=1, \dots, K}$ and eigenvalues $\{\epsilon_i\}_{i=1, \dots, K}$ as a single general eigenvalue problem. As one can imagine, various tactics for diagonalizations of the overlap and Fock matrices are employed. All of the integrals given above can be calculated as a preliminary step. Based on an initial guess for the bond-order matrix, however, the $\{C_{\mu i}\}_{\mu=1, \dots, K}^{i=1, \dots, K}$ must be iteratively computed until self-consistency with bond-order matrix is obtained—i.e., in the simplest case one is basically performing a Gauss-Seidel iteration and is subject to the corresponding Gauss-Seidel convergence criteria (Szabo and Ostlund, 1989). The resulting calculation is termed the "self-consistent field Hartree-Fock procedure."

4.3.4. Choice of Basis Function Sets

The main computational difficulty in solving the Roothaan-Hall equations rests on the large number of integrals, particularly two-electron integrals, that must be manipulated. The choice of a basis set, $\{\phi_\mu(\mathbf{r})\}_{\mu=1}^K$, is in effect a tradeoff between minimizing the total number of integrals and the difficulty of evaluating each integral in Eq. 67. For example, in order to minimize the total number of integrals, one might be tempted to use the set of Slater-type functions, which form hydrogen-like wavefunctions. The Slater-type function for the 1s-orbital centered about an atomic nucleus A , is given in Eq. 71.

$$\phi_{\text{SF},1s}(\mathbf{r}_I - \mathbf{R}_A) = \left(\frac{\zeta^3}{\pi}\right)^{1/2} \exp[-\zeta r_{IA}] \quad (71)$$

where ζ is the Slater orbital exponent, characterizing the diffuseness of the orbital. It should be noted in particular that the function does not possess a unique derivative at $r_{IA} = 0$. This discontinuity is anticipated as a qualitative feature of the molecular orbitals. A basic limitation of the Slater-type functions, however, is that they cannot be integrated analytically in the two-electron integrals. The evaluation of the integrals must be performed numerically, giving rise to a significant computational burden.

As an alternative, Gaussian-type functions allow the two-electron (four-center) integrals to be simplified to two-center integrals that are readily evaluated (Szabo and Ostlund, 1989). The 1s, 2p_x and 3d_{xy} Gaussian-type functions are given in Eqs. 72–74.

$$g_{1s}(\alpha, \mathbf{r}_I - \mathbf{R}_A) = \left(\frac{2\alpha}{\pi}\right)^{3/4} \exp[-\alpha r_{IA}^2] \quad (72)$$

$$g_{2p_x}(\alpha, \mathbf{r}_I - \mathbf{R}_A) = \left(\frac{128\alpha^5}{\pi^3}\right)^{1/4} x \exp[-\alpha r_{IA}^2] \quad (73)$$

$$g_{3d_{xy}}(\alpha, \mathbf{r}_I - \mathbf{R}_A) = \left(\frac{2048\alpha^7}{\pi^3}\right)^{1/4} xy \exp[-\alpha r_{IA}^2], \quad (74)$$

where α is the Gaussian orbital exponent and x , y are the scalar magnitudes of the Cartesian components of r_{IA} . So as to improve the functional form used, the individual basis functions are generally constructed from a linear combination of Gaussian-type functions. This construction is known as a "contracted Gaussian function" and has the form given in Eq. 75.

$$\phi_{\text{CGF},\mu}(\mathbf{r}_I, \{\mathbf{B}_p\}_{p=1}^L) = \sum_{p=1}^L d_{p\mu} g_p(\alpha_{p\mu}, \mathbf{r}_I - \mathbf{B}_p) \quad (75)$$

Although the individual Gaussian-type functions can be centered at up to L different origins, $\{\mathbf{B}_p\}_{p=1}^L$, in practice the functions are usually centered around a single atomic center. Equations of the form of Eq. 75 give rise to a number of different types of basis sets. A few are listed below.

Minimal STO-LG Basis Sets (*e.g.*, STO-3G): Minimal basis sets are those that contain the minimum number of basis functions centered on each atom required to describe the occupied atomic orbitals of the atom. STO-LG basis sets are minimal basis sets in which each basis function is a contraction of L Gaussians. The coefficients $d_{p\mu}$ and $\alpha_{p\mu}$ are fitted to approximate Slater-type functions (Szabo and Ostlund, 1989).

Split Valence Basis Sets (*e.g.*, 3-21G, 4-31G): The split valence basis sets utilize a minimal basis (as described above) for inner shell electrons and a double basis for valence electrons. The first function in the double basis consists of a contracted Gaussian function while the second is a single diffuse Gaussian. In the 3-21G basis set, each inner shell electrons are represented by a contracted Gaussian consisting of three (the 3 in 3-21G) primitive Gaussian functions. Each valence electron is represented by contracted Gaussian consisting of two (the 2 in 3-21G) primitive Gaussian functions and by a single diffuse Gaussian function (the 1 in 3-21G). It should be noted that basis sets where all of the electrons are represented by a double basis are known as double-zeta basis sets.

Polarized Basis Sets (*e.g.*, 6-31G* and 6-31G**): The polarized basis sets are essentially split valence basis sets with polarization functions added. These basis functions improve the representations of highly polarized atoms. The 6-31G* basis set augments a 6-31G split valence basis set with six d-type Cartesian Gaussian functions for each non-hydrogen atom. The 6-31G** basis set augments a 6-31G* basis set by adding three p-type Cartesian Gaussian functions.

4.4. Semi-Empirical Methods

The various semi-empirical molecular orbital methods are readily understood as simplifications of the Roothaan-Hall equations. They generally involve a combination of

- using atom-centered atomic orbitals as basis functions for the molecular orbitals;
- reducing the number of integrals to be evaluated by neglecting certain types of differential overlap (*i.e.*, certain two-electron integrals termed "exchange integrals" involving overlap of basis functions from different atomic centers);
- empirically parameterizing the electric fields generated by core electrons;

- empirically fitting certain integrals to reproduce molecular properties of interest; and
- solving the resulting equations only for the valence electrons.

Semi-empirical approaches are motivated by two main factors. First, the accurate prediction of properties by *ab initio* methods can require extremely large basis sets and multiple configuration state functions (or the use of other methods to incorporate electron correlation), possibly yielding computationally intractable problems. Second, empirical parameterization of the interaction potentials can allow quantitative accuracy to be achieved for certain properties, particularly within specific classes of compounds. In particular, because the empirical functions are derived from experimental data, it is possible to incorporate some electron correlation effects.

Currently, the most widely used general-purpose semi-empirical methods (*e.g.*, AM1: Austin Model 1; Dewar *et al.*, 1985) are based on different parameterizations of the MNDO (Modified Neglect of Diatomic Overlap; Dewar and Thiel, 1977) method, which was developed using the NDDO (Neglect of Diatomic Differential Overlap; Pople and Beveridge, 1967) approximation of the Roothaan-Hall equations. The resulting models have largely supplanted the earlier CNDO (Complete Neglect of Differential Overlap; Pople and Beveridge, 1970) and INDO (Intermediate Neglect of Differential Overlap; Pople and Beveridge, 1970) methods. The theme of these methods is the degree to which certain two electron exchange integrals (Eqs. 61 and 62) are neglected in the Fock matrix. The CNDO method effectively neglects these integrals, yielding a much simplified Fock matrix. The INDO method was designed in part to improve upon the CNDO method by partially incorporating differential overlap between pairs of single-center basis functions, but not two-center basis functions. Finally, the NDDO method incorporates a more complete treatment of differential overlap between pairs of single-center electrons.

The first simplification made by the NDDO method is to construct the wave functions only for the valence electrons. Interactions between valence electrons and the atomic core are estimated through an empirical core potential. The second simplification is to consider only those integrals involving basis functions centered on no more than two atomic centers. Of these, each two-electron integral can involve no more than two basis functions from each atom center. In effect, the types of two electron integrals included are as follows:

$$(\mu\mu, \nu\nu), (\mu\nu, \mu\nu), (\mu\mu, \lambda\sigma), \text{ and } (\mu\nu, \lambda\sigma), \quad (76)$$

where ϕ_μ and ϕ_ν are centered on an atom, call it *A*, and ϕ_λ and ϕ_σ are centered on an atom *B*. The first two integrals in Eq. 76 are the single-center Coulomb and exchange integrals for two valence electrons. Through the exchange integral the NDDO method accounts for the differential overlap between atomic orbitals on the same center, unlike the CNDO and INDO methods. The latter two integrals are diatomic repulsion integrals. In

particular, the final integral incorporates the electron repulsion between two electrons on different centers arising from differential overlap between single-center orbitals. It is important to note, however, that not included in the NDDO approximation are integrals such as $(\mu\sigma, \mu\sigma)$, which represent differential overlap between different centers, hence the designation "NDDO." A summary of the treatment of differential overlap by the different methods is given in Figure 4.1.

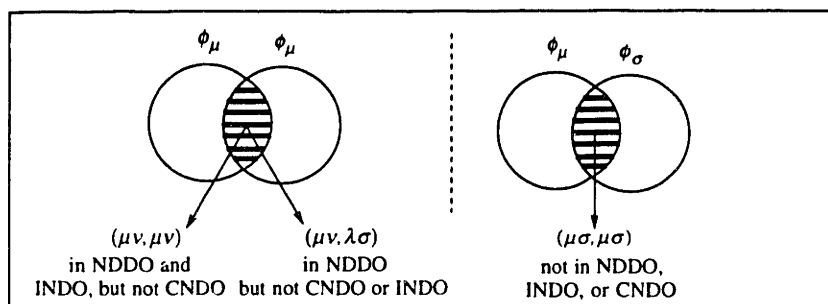


Figure 4.1. Depiction of Differential Overlap Integrals Treated by CNDO, INDO, and MNDO.

Methods such as MNDO (Dewar and Thiel, 1977), AM1 (Dewar *et al.*, 1985) and PM3 (Stewart, 1989) start with the NDDO model and introduce semi-empirical models for the core-electron attractions, core-core repulsions, one-center integrals, and two-center integrals. The parameters in these models are determined either independently (*e.g.*, the one-electron, one-center terms are independently fitted to spectroscopic measurements) or are treated as adjustable parameters by fitting the results of predicted properties to experimental measurements (as in the case of the core potential interactions and two-center integrals). The adjustable parameters in the MNDO method have been fitted to reproduce the heat of formation, ionization potential (*via* Koopman's theorem; Levine, 1991), the dipole moment, and the energy gradient. Improvements to the MNDO parameterization resulted in the AM1 parameterization, which yields better estimates of hydrogen-bond energies. A comparison of the accuracy of several semi-empirical methods is given in section six.

4.5. Estimation of Properties and Their Modeling Applications

The solution of the Roothaan-Hall equations, either through the self-consistent Hartree-Fock or semi-empirical approaches, yields the coefficients $\{C_{\mu i}\}_{\mu=1, \dots, K}^{i=1, \dots, K}$ and the energy levels $\{\epsilon_i\}_{i=1}^K$ of the molecular orbitals. This section considers several useful molecular properties that can be obtained, with particular emphasis on application in semi-empirical models and model building strategies.

Electronic Energy. The Fock matrix and bond-order matrix provide ready estimates of the total electronic energy. The ground state electronic energy is given in Eq. 77 for the Hartree-Fock approximation.

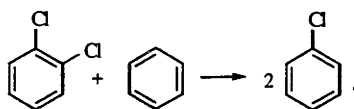
$$E_0 = \sum_{\mu=1}^K \sum_{\nu=1}^K P_{\mu\nu} (H_{\mu\nu}^{\text{core}} + F_{\mu\nu}), \quad (77)$$

In principle, the ground state energy provides a basis for prediction of molecular binding energies, bond dissociation energies, and energies for many other formal reaction processes. For example, an estimate of the binding energy of a water molecule (*i.e.*, the energy released by the reaction $2\text{H}+\text{O}\rightarrow\text{H}_2\text{O}$) can be estimated by Eq 78.

$$\Delta E_{\text{bind}} = E_0(\text{H}_2\text{O}) - E_0(\text{O}) - 2E_0(\text{H}) \quad (78)$$

It is well known, however, that the Hartree-Fock method fails to yield reliable predictions of these energies (Levine, 1991; a comparison of computational methods for different properties is presented below in the next section). The binding energy of a water molecule has been estimated in a calculation near the Hartree-Fock limit to be 6.9eV (Levine, 1991), whereas the experimental value is 10.1eV. The major shortcoming in the Hartree-Fock calculation is the failure to account for changes in the correlation energy between pairs of electrons as bonds are formed and broken. In the case of water formation, for example, a total of two new chemical bonds are formed. Significant correlation energy is associated with each bond as the two bonding electrons go from a separated, unpaired state to a paired state.

Despite the above limitation, reasonable estimates of reaction energies can be obtained for formal reactions in which the correlation effects in the reaction products effectively cancel those in the reactants. One class of such reactions are known as "isodesmic reactions," in which the total number of bonds of each type does not change as a consequence of the reaction. An obvious type of isodesmic reaction is the interconversion of certain isomers, such as the *cis* and *trans* isomers of a compound. A more general example of a formal isodesmic reaction is given by



Reaction processes of interest can often be recast as isodesmic reactions through the selection of appropriate reference compounds whose binding energies are known. In particular, reactions termed "bond separation reactions" (Levine, 1991) allow interaction

energies between different bonds in a molecule to be estimated. Take the compound $\text{CH}_3\text{-CH=C=O}$. The construction begins by forming the relevant products, each of which contains a single type of salient bond present in the molecule (Figure 4.2).

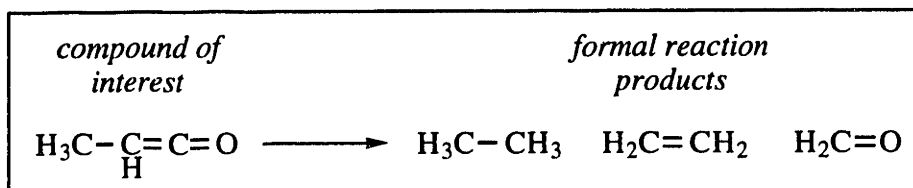


Figure 4.2. Formal Reaction Products from the Bond Separation Reaction Construction.

The isodesmic bond separation reaction is formed by balancing the reaction stoichiometry with an appropriate number of hydride molecules (*e.g.*, methane, ammonia, water, *etc.*). The formal reaction for the case of methane is given in Figure 4.3.

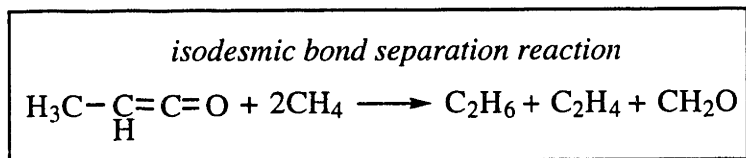
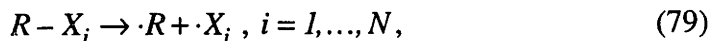


Figure 4.3. Isodesmic Bond Separation Reaction formed by Balancing Reaction Products with an Appropriate Number of Methane Molecules.

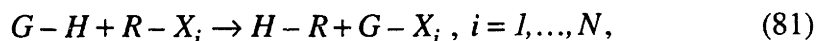
Isodesmic reactions are potentially useful in the development of certain chemical property models. Recall that statistical models do not necessarily require absolute property values, but may instead be derived from comparison of a reference reaction as in the case of linear free-energy relationships. As a simple example, consider the dissociation of a class of N compounds $\{R-X_i\}_{i=1}^N$, where X_i is the dissociating group and R is the invariant part of the molecule (Eq. 79),



with a dissociation energy given by Eq. 80.

$$\Delta E_0^i = E_0(\cdot R) + E_0(\cdot X_i) - E_0(R-X_i), \quad i = 1, \dots, N \quad (80)$$

The reactions (Eq. 79) can be converted to isodesmic reactions by adding a reactant $G-H$, to yield Eq. 81.



with a dissociation energy given by Eq. 82.

$$\Delta\tilde{E}_0^i = E_0(H - R) + E_0(G - X_i) - E_0(R - X_i) - E_0(G - H), \quad i = 1, \dots, N \quad (82)$$

The introduction of a common second reactant for the reaction series reduces the correlation error within the individual reactions. Information about the correlation energy in the original reaction, however, is lost. Thus, there is not an exact way to use Eq. 82 to estimate the original dissociation energies. However, within a sufficiently narrow class of compounds (*e.g.*, heavy hydrocarbons of different branching levels and sizes) it should be feasible to assume that the correlation interactions are almost identical for the reactions, *i.e.*,

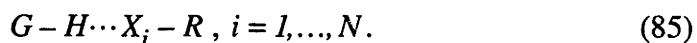
$$E_{\text{correlation}}^i = E_{\text{correlation}}^j, \quad i, j = 1, \dots, N \quad (83)$$

Thus, one might propose an empirical relation of the form given in Eq. 84.

$$\Delta E_0^i = a \cdot \Delta\tilde{E}_0^i + b, \quad i = 1, \dots, N, \quad (84)$$

where a and b are fitted parameters to be determined from a training set consisting of measured ΔE_0 values.

A good application of Eq. 84 might be the development of Polanyi-Seminov-type relationships for estimating activation energies for bond dissociation reactions. There is also evidence that formal reactions of the form of Eq. 81 could be used to estimate hydrogen-bonding energies (and charge-transfer interaction energies) of the form



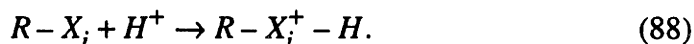
In particular, for compound series of similar geometry, reasonably accurate correlations of the form

$$-\Delta H_{HB} = a \Delta PA + b \exp[c|\Delta PA|] \quad (86)$$

have been found (Zeegers-Huyskens and Huyskens, 1991) to apply for pairs of species in polar aprotic solvents. The term ΔH_{HB} is the enthalpy of the hydrogen bond and ΔPA is the difference in proton affinities between the two species in Eq. 85. The proton affinities are defined as the negative of the molar enthalpies for the following two gas phase reactions (Eqs. 87 and 88).



and



The model fits obtained are comparable in accuracy to *ab initio* gas phase calculations involving large basis sets and with treatments of correlation. Thus, under reasonable assumptions it may be feasible to develop correlations of the form

$$\Delta PA^i = a \cdot \Delta \bar{E}_0^i + b, \quad i = 1, \dots, N. \quad (89)$$

Ionization Potential. A second property of interest is the first ionization potential, which is the energy required to remove an electron from a molecule:



The ionization potential can be estimated from open shell molecular orbital calculations. More often, however, the ionization energy is estimated as the negative of the energy eigenvalue for the highest occupied molecular orbital (*i.e.*, Koopmans' theorem; Levine, 1991).

The ionization potential has been a criterion to judge reaction feasibility in certain free radical reactions. For example, Nishino *et al.* (1989) developed an ionization potential criterion for substrate feasibility (highly activated aromatic compounds) in Manganese(III) acetate ($[\text{Mn}_3\text{O}(\text{AcO})_6(\text{OAc})(\text{HOAc})] \cdot 5\text{H}_2\text{O}$) catalyzed formylation in the presence of malonic acid, $\text{CH}_2(\text{CO}_2\text{H})_2$. An effective upper bound of 7.4 eV on the ionization potential of aromatic substrates has been established, above which the reacting substrate tends to oxidize to the carboxylic acid.

Electronic Density. A third molecular property that has received considerable attention in model building is the electron density. The electron density surrounding a molecule at a point \mathbf{r} is estimated directly from the Hartree-Fock wavefunction as $\Psi_0^*(\mathbf{r})\Psi_0(\mathbf{r})$, or in terms of the bond-order matrix as

$$\rho(\mathbf{r}) = \sum_{\mu=1}^K \sum_{\nu=1}^K P_{\mu\nu} \phi_{\mu}^*(\mathbf{r}) \phi_{\nu}(\mathbf{r}). \quad (91)$$

For reasons discussed below, the electron density can be estimated by the self-consistent Hartree-Fock method. As a result, it has been a source of considerable interest for development of property models. For example, Politzer and workers have published a series of papers in which the electron density is used in a set of empirical descriptors to

correlate thermophysical properties such as the normal boiling point and partition coefficients of organic compounds. A generating function for the descriptors is the electrostatic potential (Politzer and Truhlar, 1981), defined by Eq. 92.

$$V^{\text{ES}}(\mathbf{r}) = \sum_{A=1}^{n_A} \frac{Z_A}{|\mathbf{r} - \mathbf{R}_A|} - \int \frac{\rho(\mathbf{r}_1) d\mathbf{r}_1}{|\mathbf{r} - \mathbf{r}_1|}. \quad (92)$$

The electrostatic potential is the predicted potential field generated by an isolated molecule in a vacuum and has been used in development of empirical models for prediction of thermodynamic properties, non-covalently bonded interactions, and reaction barriers. For example, Kollman (1981) has discussed models for the form

$$\Delta E = \text{constant} \cdot V^{\text{ES}}(\mathbf{R}_{LA}) V^{\text{ES}}(\mathbf{R}_{LB}) \quad (93)$$

for estimation of Lewis acid (*LA*)-Lewis base (*LB*) interaction energies, where \mathbf{R}_{LA} and \mathbf{R}_{LB} are reference positions for a series of compounds (*i.e.*, two coordinates associated with some configuration of the two compounds are somehow chosen). The form of Eq. 93 has been justified by analogy to the formulas for interaction energies between point charges and point dipoles.

As a second example, Tomasi (1981) has developed correlations for activation barriers in protonation on the basis of a more rigorous form of Eq. 93, namely the electrostatic potential field generated by two molecules (call them "1" and "2"),

$$E_{\text{ES}} = \int V_1^{\text{ES}}(\mathbf{r} - \mathbf{R}_1) \rho_2(\mathbf{r} - \mathbf{R}_2) d\mathbf{r}. \quad (94)$$

where, again, \mathbf{R}_1 and \mathbf{R}_2 are orienting vectors that must be determined somehow.

Although the electronic density is a continuous distribution, it is often convenient to partition the charge density into specific atom-centered contributions. There is no unique way of making such an assignment and several approaches are in common use. The most widely used method is Mulliken population analysis. In Mulliken population analysis, an atom center, call it *A*, contains all of the electron density centered on it and one-half the electron density shared with other atoms. The formula is given in Eq. 95.

$$q_A = Z_A - \sum_{\mu \in A} \sum_{a=1}^K P_{\mu a} S_{a\mu} \quad (95)$$

Partial charges have been used as measures of electron density in several types of predictive models. In the next chapter, the use of partial charges as scaling factors in pure

component property prediction (Grigoras,1991) is discussed. Partial charges are also used in linear free energy models of the form

$$\log \frac{k}{k_0} = \rho(k_0) \cdot \sigma(k), \quad (96)$$

where k is the reaction rate for the substrate of interest and k_0 is the rate for a reference reaction characteristic of the reaction class (Neurock *et al.*, 1989). The reaction class is characterized by a reaction constant $\rho(k_0)$ for specific reaction conditions and the substituent variability by $\sigma(k)$, which is ideally independent of reaction conditions. It has been asserted that the substituent constant, $\sigma(k)$, is proportional to various measures (*e.g.*, the π contribution) of the partial atomic charge on the reacting substrate.

Mullikin population analysis has been criticized on the basis that it tends to yield excessively high charge separation, particularly for light atoms such as hydrogen. An alternate strategy for assigning atomic charges is electrostatic potential fitting, wherein charges are assigned by a fitting procedure to approximate the electric field generated by the density function. It should be noted that such a definition of partial charges probably might not work well for a number of property models (*e.g.*, linear free-energy relationships) that rely on reference compounds.

4.6. Comparison of Computational Methods

Considerable computational experience in property prediction with different basis sets has been reported in the literature. General conclusions as to the scope of the Hartree-Fock method (Levine, 1991) are that

- self-consistent field Hartree-Fock calculations are reasonably good in the prediction of properties that involve one-electron calculations (*e.g.*, electron density at the optimal geometry and the dipole moment);
- for such properties, the accuracy of the prediction is improved through larger, more complete basis sets without necessarily requiring methods that predict the correlation energy; and
- self-consistent field Hartree-Fock calculations are not adequate for most properties that depend strongly on two-electron integrals to the electron correlation effects (*e.g.*, absolute energies, non-isodesmic reaction energies).

A summary comparison of the choice of basis functions and estimation methods for different types of properties has been given by Levine (1991). This section summarizes Levine's comments for a few properties of interest.

4.6.1. Molecular Geometry

Molecular geometries (bond distances, bond angles, and dihedral angles) are in general predicted well by minimal basis self-consistent field Hartree-Fock calculations, with consistent improvements for the series STO-3G, 3-21G, 3-21G*, and 6-31G*. Absolute errors for bond lengths are on the order of 0.03\AA and bond angles on the order of 3° . The accuracy tends to be lower for second row atoms, as might be expected due to greater polarizability and diffuseness.

Bond distances and angles are predicted well by the semi-empirical methods MNDO and AM1. Dihedral angles, however, are not predicted accurately.

4.6.2. Absolute Energies and Energies of Reaction

Absolute atomic and molecular energies predicted by the Hartree-Fock method tend to exhibit errors on the order of 0.5–1%. As Levine (1991) points out, the error is small in absolute terms but is sufficiently large to make chemical property predictions unreliable. For example, the total energy of the carbon atom is *ca.* 1000 eV. An 0.5% error in this value runs 5 eV, which is the same order of magnitude as bond energies. Although improved values can be obtained (up to a limiting value known as the "Hartree-Fock limit") through larger basis function sets, quantitatively accurate absolute energies require estimation of the correlation energy. Significantly better energies can be obtained through the use of configuration interaction methods or other techniques, such as the Møller-Plesset perturbation theory (Szabo and Ostlund, 1989). Alternately, semi-empirical methods such as AM1 tend to give reasonable estimates since they are parameterized from the molecules' enthalpy of formation. It should be remembered, however, that semi-empirical methods tend to be limited by the set of molecules used to parameterize the underlying models. Large deviations can be expected for many of the molecules outside the data set.

4.6.3. Ionization Energies

Related to the absolute energy, the ionization energy can be estimated as the negative of the energy of the highest occupied molecular orbital (a result termed Koopmans' theorem; Levine, 1991). *Ab initio* methods generally yield ionization potentials within 1–1.5 eV of measured values. The error can be reduced significantly through larger basis sets (*e.g.*, 6-31G*). MNDO and AM1 methods also tend to be relatively accurate, typically within 0.5–0.8 eV (Levine, 1991).

4.6.4. Dipole Moments

Dipole and higher moments can be estimated directly from the density function. The dipole moment estimated from the Hartree-Fock approximation is given by Eqs. 97 and 98.

$$\bar{\mu} = \int \Psi_0^*(\mathbf{r}) \mathbf{r} \Psi_0(\mathbf{r}) d\mathbf{r} + \sum_{A=1}^{n_A} Z_A \mathbf{R}_A \quad (97)$$

or, to within the accuracy of the basis set, by

$$\bar{\mu} = \sum_{\mu=1}^K \sum_{\nu=1}^K \int \phi_{\mu}^*(\mathbf{r}) \mathbf{r} \phi_{\nu}(\mathbf{r}) d\mathbf{r} + \sum_{A=1}^{n_A} Z_A \mathbf{R}_A \quad (98)$$

The dipole moments predicted by minimal basis sets tend to be significantly in error. For example, a comparison done on 21 small molecules (Levine, 1991) yielded absolute error on the order of 0.6 D (D [=] Debye). Split valence basis sets reduce the error somewhat to levels on the order of 0.4 D. Significantly improved estimates (say 0.3 D) can be obtained by incorporating split valence basis sets with polarization functions (*e.g.*, 6-31G*). It should be noted that CNDO and INDO methods are not reliable at all. Semi-empirical methods such as MNDO, AM1, and PM3 yield average absolute errors on the order of 0.4 D or more.

4.7. Discussion

A number of molecular properties have been identified that could be incorporated into model-building strategies for the prediction of compound properties. These include absolute energies, ionization energies, molecular geometry parameters, electron density parameters, and reaction energies. The applicability of these parameters in models must be balanced against the computational requirements. For example, the prediction of accurate absolute energies can involve significant computational resources for even moderate sized molecules, say with molecular weights on the order of 100. If hundreds or thousands of compounds are to be screened on a routine basis, then perhaps strategies for exploiting isodesmic reaction series should be considered.

A second consideration is the ability of these parameters to reduce the number of adjustable fitted parameters as compared to existing models which do not exploit quantum calculations. A prime example are group contribution thermodynamic property models which incorporate geometric parameters for different functional groups. Such models, for example, define different geometric descriptors and property contribution parameters for CH₃, CH₂, CH and C. Thus, four groups are required to characterize only two atoms of a single hybridization state. Geometry optimizations followed by numerical prediction of the molecular volume or surface areas should allow all these groups to be represented by appropriate combinations of two fundamental groups: C_{sp³} and Haliphatic.

A third consideration is the statistical significance of the parameters in the property models. Each parameter described above has been shown to be significant in a particular

model, but the development of transferable parameters should require considerable experimentation with different parameters and model forms. In particular, it is useful to consider parameters that can be defined at different levels of rigor and the statistical significance of refinements tested. Such a parameter is the electron density, which can be characterized in several ways, ranging from simple partial atomic charges to explicit density descriptors (Politzer, 1993). It is important to keep in mind, however, that the parameters will most certainly have to be mixed with fitted parameters to obtain good accuracy. For example, the statistical parameters defined by Politzer (1993), which involve essentially no fitted parameters, are important scientifically but do not give accurate prediction of normal boiling points say, within 15 K.

With the above considerations in mind, the starting point for model development considered in the next two chapters is to exploit the semi-empirical AM1 method for estimation of molecular geometries and Mullikin partial charges. The rationale is as follows:

- the AM1 method provides rapid solution times and can therefore be automated (say within the retrosynthetic analysis code) for large numbers of compounds;
- the partial charge parameters are already used in a wide range of models;
- the limitations and requirements for improving the partial charges (*e.g.*, electrostatic potential fitting, dipole constraints) are well understood, can be refined, and the statistical significance tested; and
- by working with partial atomic charges, the resulting model forms should be reasonably simple so that good statistical analysis can be performed.

Two concerns associated with the AM1 method are the accuracy of the partial charges (particularly for second row atoms), which has been addressed in the rationale, and the accuracy of dihedral angles. It will turn out, however, that errors in the predicted dihedral angles will have little effect on the results.

4.8. Conclusions

The non-relativistic time-independent Schrödinger wave equation has been introduced as the governing equation for many molecular electronic structure problems. As a starting point for the solution of this equation, the Hartree-Fock approximation has been presented. The main limitation of the Hartree-Fock treatment is the failure to account for electron-electron correlation. As a result, the method fails to accurately predict absolute energies. However, the method in fact underlies most current methods, including those that estimate electron-electron correlation. Semi-empirical methods have also been introduced. A discussion of molecular properties with an emphasis on model building has been given. It is concluded that with currently available computational power, there is a strong motivation for exploiting quantum chemical calculations in model development. The Hartree-Fock and

semi-empirical methods in particular should be useful in models that involve geometric and electron density parameters.

Chapter 5

Integrated Model Building Strategy Incorporating Molecular Calculations. Computational Tools.

5.1. Introduction

With the growing accessibility and widespread interest in quantum calculations, there is a strong motivation for applications development tools for these methods. The number of public domain and commercial codes for general purpose molecular orbital applications has increased considerably in recent years (*e.g.*, MOPAC 93 (Stewart, 1993) and GAUSSIAN 92 (Trucks *et al.*, 1992)). These codes contain a hierarchy of computational methods ranging from semi-empirical methods, such as AM1 (Dewar *et al.*, 1985), to *ab-initio* methods with a variety of available basis functions. Computational options include single point calculations, geometry minimizations, transition state optimizations, reaction field methods, and excited state calculations.

Quantum chemistry is no longer confined to the domain of pure research and theory. Increased computer power, more interactive operating systems, and improved visualization capabilities have catalyzed interest in quantum chemistry as an integral part of research and development. Networked engineering workstations are now available in many research and development centers alongside the laboratory facilities. As a result, a range of computational models for prediction of chemical properties are currently being pursued by researchers. An important class of these models are those that combine theory and experimental measurement, thereby seeking to achieve good accuracy for specific classes of compounds. Of particular interest to us are prediction of pure component

properties as a basis for extension to mixture properties. While both pure component and mixture properties are described by existing statistical mechanical theories, incorporation of molecular computations into predictive models could significantly improve the accuracy and robustness of the models. Better prediction of molecular properties could potentially improve laboratory screening, reduce experimental costs, and lead to new insights.

Modeling requires integration of molecular calculations with model compound data, parameter fitting tools, and relevant theory for the properties of interest. Applications development software should possess the following attributes:

- an interface to a computation chemistry engine;
- properties databases;
- model specification tools;
- correlation and statistical analysis tools; and
- an interface to applications software.

This chapter presents a property modeling architecture for the integration of molecular surface area (and volume) and electronic density measures. A key precedent to this work is the structural approach due to Grigoras (1990). Whereas Grigoras dealt with models essentially linear in the group energy contributions, this work proposes a more general framework. The framework exploits the molecular surface area as a measure of intermolecular coordination, allowing for empirical models that explicitly account for atom-atom interaction. An impetus for the approach is the use of molecular surface area concepts in group contribution activity coefficient models such as UNIFAC (Fredenslund *et al.*, 1975). The molecular surface area concept originates from statistical lattice theories and is essential to describing the coordination of a solvent around a polysegmented molecule. Therefore, incorporating molecular surface area measures appears to be a good starting point for model development. A second consideration is that traditional group contribution methods tend to average out group differences in the statistical parameter fitting process. The electron density and other measures derived from molecular orbital calculations can potentially incorporate these structure-sensitive dependencies in group contribution models.

The computational framework presented below is capable of (i) building, evaluating, and testing the class of models described above; and (ii) incorporating property prediction calculations within applications programs. The computational tools developed are as follows:

- tools for specification and generation of large data sets of molecular calculations in GAUSSIAN 92;
- a hybrid parameter optimization strategy using both linear regression and the Levenberg-Marquardt algorithm for the synthesis of a class of property models; and

- a Monte Carlo integration code for efficient determination of molecular surface area measures.

5.2. Overview

A computational framework is proposed for synthesis of a class of property models, among them pure component properties and linear solvation-energy relationships. Exploited are molecular surface area calculations, which provide a statistical thermodynamic basis for coordination of molecules in condensed systems, and electronic density calculations, which incorporate structural information traditionally averaged out by statistical group contribution methods. The computational framework is tailored to nonlinear models involving atom-atom interaction types (*e.g.*, dispersion, polar, and hydrogen bonding). These interactions are scaled by the molecular surface contributions to reflect intermolecular coordination using electronic density measures. A two-stage nonlinear regression strategy is employed for parameter optimization. The Levenberg-Marquardt algorithm is used for optimization of empirical atom-atom interaction parameters, while linear regression is used at each Marquardt iteration for updating linear multipliers of the interactions. Molecular surface area measures are computed with a parallelizable Monte Carlo integration strategy that uses shifted-Hammersely sample points. Having determined a property model, computational tools for property prediction are available for incorporation within applications programs.

5.3. Chemical Property Modeling

Linear relationships between properties of interest and the intermolecular potential play an important role in semi-empirical model development, particularly in condensed phase systems. In practice, the intermolecular potential is often separated into weighted contributions of dispersion, polar, and hydrogen bonding terms. The resulting semi-empirical models take the form given in Eq. 1.

$$F^m = \sum_{n=1}^N \alpha_n f_n[m], \quad (1)$$

where F^m is the property of interest for a compound, m , and the $f_n[\bullet]$ ($n = 1, \dots, N$) are functionals defining the overall intermolecular potential contributions. The α_n are property-specific weighting parameters which ideally incorporate thermodynamic characteristics common to the class of molecules being modeled. Such weighting parameters can be formally justified through assumptions related to the combinatorial entropy. The class of properties that can be treated by Eq. 1 include those that are amenable to a linear free-energy assumption—*i.e.*, those in which the combinatorial entropy contribution can be estimated through measurements for a reference compound. Examples include the normal boiling points, critical temperatures, and critical volumes of

many organic liquids, as well as certain mixture and reactivity properties such as linear solvation-energy relationships (derivations for these properties are given in chapter six).

Whereas the α_n ($n = 1, \dots, N$) are specific to a particular property, the $f_n[\bullet]$ are intrinsic descriptors of the intermolecular potential that can, in principle, be transferred to other properties that depend on the same intermolecular potential. The dichotomy motivates a model building strategy in which properties in abundance, such as pure component properties, are utilized as a basis for developing transferable interaction potential parameters. Based on the form of Eq. 1, these parameters can be separated from the weighting coefficients and used to estimate additional properties. A general model building strategy utilizing this idea is depicted in Figure 5.1.

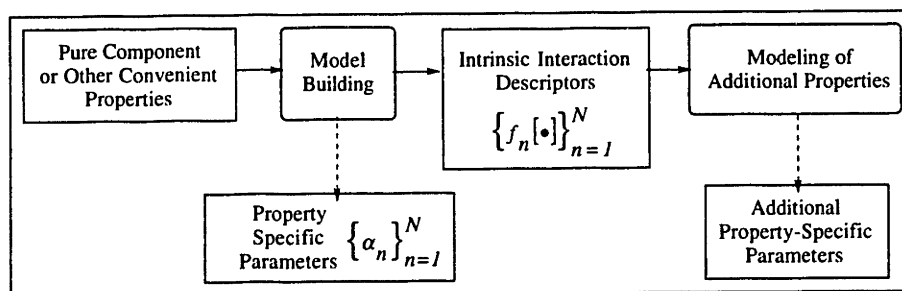


Figure 5.1. Model Building Strategy.

Figure 5.1 is a superstructure of several strategies in current use. Group contribution methods are frequently employed to derive parameters for common functional groups that can then be transferred to a large number of molecules outside the original training set. Group contribution methods can be either additive or nonlinear and have been applied to pure component properties, mixture properties, and reactivity (*e.g.*, Hammett's relationship). A second strategy is to use measured properties to form empirical relations for the descriptor of interest. A good example is the set of solvachromatic parameters used in linear solvation-energy relationships (Reichardt, 1990).

While these and other approaches have been successful for many problems, they can be difficult to implement in industrial practice. First, the number of parameters required to obtain accurate correlations can grow quite large. In group contribution methods, for example, differences in functional groups (*e.g.*, electronic density) that can lead to different properties tend to average out during statistical fitting. Such averaging can be deleterious when one is interested in developing good models for a particular class of compounds. Furthermore, refinements to the group contributions may introduce additional parameters, thus increasing the number of model experiments (*i.e.*, the laboratory cost) required to obtain good statistical accuracy. Second, the requirement of direct property measurements such as solvachromatic shifts may be tedious and expensive if one is concerned with screening a large number of new compounds. In both cases, while

parameters may be both refinable and transferable in principle, in practice the experimental costs could be limiting.

5.4. Incorporation of Molecular Calculations

Several proposals have been advanced recently on the incorporation of molecular calculations within semi-empirical models for estimating thermophysical properties. Of particular interest has been the incorporation molecular surface area and electronic density parameters. Grigoras (1990) has proposed functional forms that are essentially linear in the fitted parameters. Three molecular surface descriptors have been proposed for the dispersion interaction and two types of polar interactions (a fourth, hydrogen bonding function of a similar form has also been proposed). For a molecule m consisting of a set C_m of atom-types—*i.e.*, atoms in different hybridization states—the descriptors are given by Eqs. 2–4.

$$\text{Dispersion Function : } f_1(m) \equiv A_{disp}(m) = \sum_{c \in C_m} A_c(m) \quad (2)$$

$$\text{Electrostatic Negative Function : } f_2(m) \equiv A_-(m) = \sum_{\{c \in C_m \mid q_c(m) < 0\}} b_c q_c(m) A_c(m) \quad (3)$$

$$\text{Electrostatic Positive Function : } f_3(m) \equiv A_+(m) = \sum_{\{c \in C_m \mid q_c(m) > 0\}} b_c q_c(m) A_c(m) \quad (4)$$

where: $A_c(m)$ is the exposed molecular surface area of component $c \in C_m$, $C_m \subseteq C$ (the complete set of atom-types); $q_c(m)$ is the atomic group charge determined from a semi-empirical molecular orbital calculation; and b_c is a charge scaling factor. Several linear correlations of physical properties have been developed using Eqs. 2–4, including the critical volume, critical temperature, and normal boiling point. The quality of the correlations is generally comparable to traditional group contribution methods such as Joback's method (Ried *et al.*, 1987).

Equations 2–4 can be rationalized in part by interpreting the molecular surface area as a metric for the degree of coordination between atoms on different molecules. In the classical theory, the molecular surface area is proportional to the number of solvent coordination bonds around a molecule (Staverman, 1950). Hence, for molecules containing similar functional groups, a larger surface area should result in larger [orientation independent] dispersive and polar interactions. This attribute is qualitatively present in Eqs. 2–4.

Whereas in group contribution methods a limited set of functional groups are assigned structure-independent surface area parameters, the surface areas are computed

directly from atom-type contributions in the approach of Grigoras. Such an approach has several advantages:

- the number of parameters is reduced because one is dealing with atom-types instead of functional groups, which involve permutations of atoms;
- it is easier to incorporate new functional groups in a model since one does not need to determine surface areas for the new groups;
- the predicted surface areas should be more accurate since they are based on the entire molecular structure; and
- better estimates of surface areas should lead to more accurate parameters within the descriptors.

Less clear in the method of Grigoras (1990) is the statistical significance of the partial atomic charges. One test of statistical significance that comes to mind is the following: how does the model compare to a second one in which the charge scaling factors are refit with all of the partial charges set to unity?

Politzer has adopted an approach in which the electron density is shown to be a highly significant statistical parameter (Politzer and Truhlar, 1981). The electron density is used to define a set of interaction descriptors with essentially no adjustable parameters. The generating function for the descriptors is the electrostatic potential (Politzer and Truhlar, 1981), defined by Eq. 5 as the predicted electrostatic field generated by an isolated molecule in a vacuum.

$$V^{\text{ES}}(\mathbf{r}) = \sum_{A=1}^{n_A} \frac{Z_A}{|\mathbf{r} - \mathbf{R}_A|} - \int \frac{\rho(\mathbf{r}_j) d\mathbf{r}_j}{|\mathbf{r} - \mathbf{r}_j|}, \quad (5)$$

where $\rho(\mathbf{r})$ is the Hartree-Fock electron density at a position coordinate, \mathbf{r} , Z_A is the nuclear charge, and \mathbf{R}_A is the position of an atomic center, A . Several statistical metrics are defined from the electrostatic potential, including a charge separation metric (Eq. 6), an electrostatic interaction tendency (Eq. 7), and an electrostatic balance term (Eq. 8) (Murray *et al.*, 1993).

$$\Pi = \frac{1}{n} \sum_{i=1}^n |V^{\text{ES}}(\mathbf{r}_i) - \bar{V}_s^{\text{ES}}|, \quad (6)$$

$$\sigma_{\text{tot}}^2 = \sigma_+^2 + \sigma_-^2 = \frac{1}{m} \sum_{i=1}^n [V^+(\mathbf{r}_i) - \bar{V}_s^+]^2 + \frac{1}{n} \sum_{j=1}^n [V^-(\mathbf{r}_j) - \bar{V}_s^-]^2, \quad (7)$$

$$v = \frac{\sigma_+^2 \sigma_-^2}{[\sigma_{\text{tot}}^2]^2}, \quad (8)$$

In Eqs. 6–8, the indices i and j refer to sample points drawn from the electrostatic potential at different positions. $V^+(r_i)$ is the positive contribution to the electrostatic potential at a point i (set to zero at points where the electrostatic potential is negative) and \bar{V}_s^+ is the averaged positive electrostatic potential.

The models developed using Eqs. 6–8 include linear combinations of the descriptors. In some cases, new descriptors are formed from exponentiation and multiplication of two of the descriptors. Given the relatively small number of adjustable parameters, the approach has been very successful in reproducing thermophysical properties, although it yields larger variances than group contribution methods for certain pure component properties. While the underlying descriptors are essentially property-independent, it is difficult to see how one would refine the descriptors. In particular, it would be interesting to see whether increases in the quality of the calculation lead to improved property estimates.

5.5. New Strategy for Industrial Model Development

Two desirable attributes of a modeling strategy are the ability to refine and to transfer model parameters to other models. While the methods discussed above have embodied one aspect or the other, there is clearly a need to develop methods which target both attributes. The strategy proposed in this chapter is to develop an explicit formula for the intermolecular potential in terms of atom-atom interactions *via* a formal lattice potential. A key motivation is the ability to use statistical mechanics to derive functional forms and to quantify assumptions, particularly those related to the combinatorial entropy. In the long run, the approach should allow the lattice potential to be incorporated within more complex property models, such as activity coefficient models. Such an extension requires a knowledge of the statistical partition function to determine the combinatorial entropy.

Formulation of the lattice potential begins with a model for the coordination between different molecules. For the athermal case, which is a first approximation, a suitable relationship for the degree of coordination between two components is given by Eq. 9.

$$\theta_{c_1 c_2}(m) = p_{c_1}(m) p_{c_2}(m), \quad c_1, c_2 \in C_m \quad (9)$$

where

$$p_c(m) = A_c(m) / A(m), \quad c \in C_m, \text{ and} \quad (10)$$

$$A(m) = \sum_{c \in C_m} A_c(m). \quad (11)$$

Eq. 9 is essentially the joint probability of two components in two non-interacting compounds being coordinated to one another. Thus, the total number of coordination bonds in a cell between a component c_1 and c_2 is given by

$$z A(m) \theta_{c_1 c_2}(m), \quad (12)$$

where z is the coordination number of the lattice. The intermolecular potential between the two components in the cell can be represented by Eq. 13.

$$U_{c_1 c_2} = z A(m) u_{c_1 c_2} \theta_{c_1 c_2}(m), \quad (13)$$

where $u_{c_1 c_2}$ represents the intrinsic intermolecular potential between the two components.

The basic strategy of this work is to use a coordination model of the form of Eq. 13 to generate a class of functional forms, $f_n[\bullet]$, for different component-component interactions. Of special interest is to see if it is possible to use an electron density metric as an empirical parameter to approximately separate $u_{c_1 c_2}$ into single component parameters, thus reducing their number considerably. The resulting component-component interactions should take the following form:

$$U_{c_1 c_2} = z A(m) \left\{ u_{c_1} p_{c_1}(m) \right\} \cdot \left\{ u_{c_2} p_{c_2}(m) \right\}, \quad (14)$$

where the product $u_{c_1} \cdot u_{c_2}$ is the ideal binary interaction parameter between two components in isolation. Several desirable properties can be inferred from Eq. 14. First, the interaction between components on two identical molecules should be symmetric to interchange in the order of arguments in $f_n[\bullet]$. Second, it should be possible to incorporate binary parameters, where needed, by restricting the scope of some of the $f_n[\bullet]$ to specific interactions. More generally, it may be desirable to include or exclude specific atom-types from a particular $f_n[\bullet]$.

A general form of $f_n[\bullet]$ satisfying these requirements is given in Eq. 15.

$$\begin{aligned}
 & f_n \left[m : \left\{ \mathit{par}_c, \mathbf{M}_c^m \right\}_{c \in C_m \cap C_n} \right] = \\
 & = A(m) \cdot \sum_{c_1} \sum_{c_2} \varphi_n \left[\mathit{par}_{c_1}, \mathbf{M}_{c_1}^m \right] \varphi_n \left[\mathit{par}_{c_2}, \mathbf{M}_{c_2}^m \right], \quad c_1, c_2 \in C_m \cap C_n, \quad (15)
 \end{aligned}$$

where C_n is the set of components within the scope of $f_n[\bullet]$. The molecular information associated with each component, $c \in C$, is defined by the vectors par_c and \mathbf{M}_c^m (note: to simplify the notation, we do not represent the individual components by a separate index from the list of component types; it is implicitly understood that a molecule can have multiple components of the same component-type). The vector \mathbf{M}_c^m characterizes the state of a component $c \in C$ within a compound m and is defined by Eq. 16.

$$\mathbf{M}_c^m \equiv \langle \mathit{area}_c^m, \mathit{den}_c^m \rangle \quad (16)$$

where area_c^m is the vector of molecular surface measures and den_c^m is the vector of electronic density measures for the component. The vector par_c is the vector of fitted parameters within the functionals $f_n[\bullet]$. For Grigoras' approach (Grigoras, 1990), $\mathit{par}_c \equiv \langle b_c \rangle$ and $\mathbf{M}_c^m \equiv \langle A_c(m), q_c(m) \rangle$ (although the functions proposed by Grigoras do not satisfy Eq. 10). Each function $\varphi_n[\bullet]$ has a general form defined by Eq. 17,

$$\varphi_n : (\mathit{par}, \mathbf{M}) \rightarrow \text{scalar}, \quad (17)$$

wherein the input vectors \mathbf{M} and par are dummy variables for the input vectors from any of the $c \in C_n$ components.

5.6. Computational Tools

The computational tools required to build models of the combined forms of Eqs. 1 and 15 can be illustrated by considering the component pieces of the property prediction problem (Figure 5.2). Given the intermolecular potential descriptors, the weighting coefficients are to be determined by linear regression for the property of interest. Within the descriptors, the fitted parameters are to be determined by nonlinear regression of a reference property(ies) for a set of model compounds. Since the weighting coefficients must also be determined, a strategy is employed in which the optimization of the weight coefficients is treated as an "outer problem" and the nonlinear regression as an "inner problem."

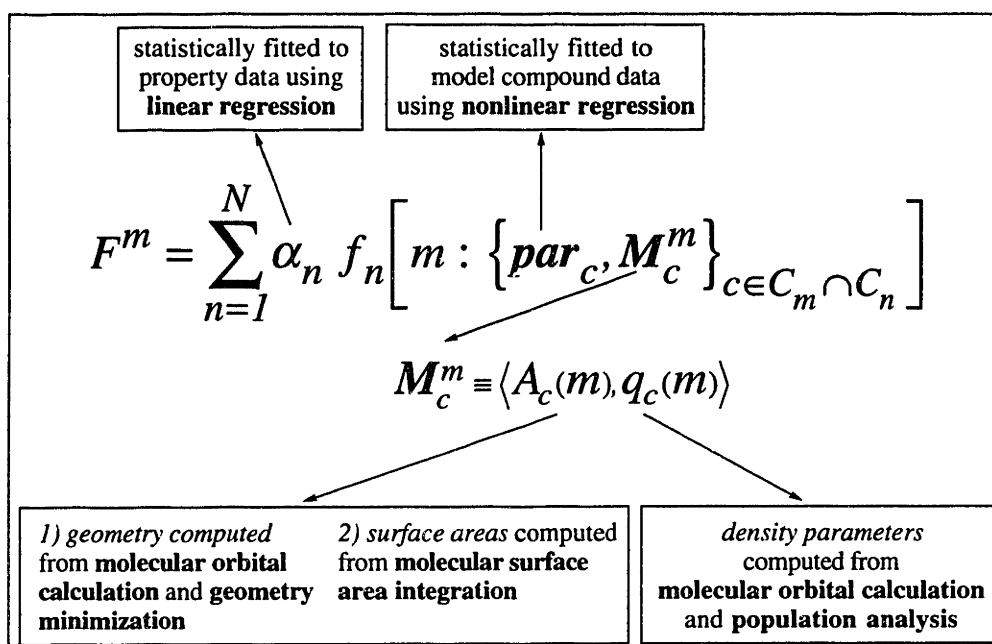


Figure 5.2. Details of Model Building Strategy.

The M_c^m are determined from molecular calculations. Molecular orbital calculations are used to optimize the molecular geometry (*i.e.*, find the minimum energy structure) and to generate the corresponding electron density. Population analysis is used to determine approximate partial atomic charges. Given the molecular geometry and a set of atomic radii, surface areas are computed using a Monte Carlo integration strategy.

This chapter develops a set of efficient computational tools to allow the development of models of the forms defined by Eqs. 1 and 15. Three main activities are pursued in the development of the tools. A prerequisite for the model building activities is the management of datasets of molecular calculations, *i.e.*, generating initial molecular structures and input files to molecular codes and retrieval of computational results. The model building effort itself requires statistical modeling capabilities for testing models and fitting parameters. A number of components developed in the first two stages are then useful in property prediction problems. The following section presents an overview of the computational architecture.

5.7. Specification, Execution, and Archiving of Molecular Orbital Calculations

Molecular orbital calculations have been performed on approximately 160 organic compounds during the research. The set of calculations has arisen in two main applications. First, approximately 145 of the compounds have been used in the

development and testing of models. Second, additional compound calculations have arisen directly from applications programs such as the retrosynthetic analysis code (chapter nine).

The need to efficiently manage such a large number of calculations has led to development of user-specified and applications-driven computational tools for the specification, execution, and archiving of multiple compound calculations. The computational architecture of the tools is depicted in Figure 5.3.

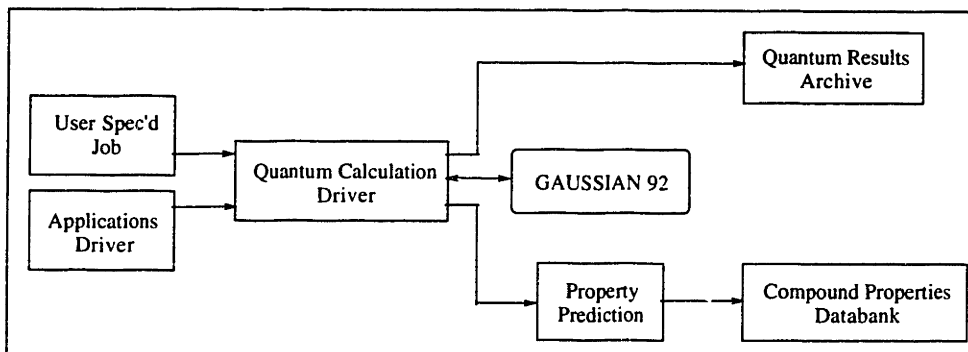


Figure 5.3. Computational Architecture for the Specification, Execution, and Archiving of Molecular Orbital Calculations in the Cray C90 Environment.

User-specified quantum calculations are responsible for generating quantum calculation datasets used for model building. According to the desired computational specifications, a set of GAUSSIAN 92 input files containing the molecular structures of interest are generated and the computations executed and returned. The results of these calculations are archived for use in model building.

5.7.1. Quantum Calculations Databank

Appendix one contains a listing of 159 compounds whose AM1 geometry optimized structures are now readily available for further model building applications. The appendix lists the compound, output file name, pure component properties, and computed molecular descriptors (based on calculations performed in chapter six). The databank includes a wide range of organic compounds with normal boiling points in the range *ca.* 200 – 600K and molecular weights *ca.* 50 – 200. Table 5.1 lists the main categories of compounds present in the databank.

Table 5.1. List of Main Compound Classes and Compound Databank Entries Used in Model Development.

Compound Class	No. Compounds	Compound Databank Entries
hydrocarbons	23	1, 7, 9–11, 21–29, 142–147, 159–161
chlorinated hydrocarbons	12	6, 15–17, 31–38
amines	17	3, 12–14, 41–48, 54–56, 64–65, 69
nitriles	6	5, 51–53, 63, 69
nitro-compounds	7	2, 60–62, 66–68
OCO-containing*	28	71–98
alcohols	26	4, 18–20, 101–114, 116–123
aldehydes	8	131–137, 139
ethers	8	151–158

*carboxylic acids, esters, etc.

5.8. Model Building

5.8.1. Computation of Molecular Surface Area

A key requirement in the chemical modeling strategy is the computation of molecular surface area. As stated above, the magnitude of the molecular surface area determines the relative degree of coordination of a molecule, *i.e.*, a molecule with a larger surface area has more molecules around it. The reasoning also applies to the individual atom-types that constitute the molecule. With regard to its surface area, a molecule consisting of a set of components, $c \in C_m$, can be thought of as a set of overlapping spheres. The number of coordinating components in surrounding molecules depends on the *exposed* surface area, $A_c(m)$, of a particular component. Thus, for example, the value of $A_{sp^3 \text{ carbon}}$ in a methyl group is certainly larger as compared to the central carbon in a *tert*-butyl group. The exposed surface area is a function of the geometry of the specific molecule as well as the radii of each component in the molecule. In general,

$$A_c(m) = A_c \left[m : \{r_c, \mathbf{x}_c(m)\}_{c \in C_m} \right], \quad (18)$$

where r_c is the radius and $\mathbf{x}_c(m)$ is the vector defining the atomic coordinate of $c \in C_m$.

Several measures of the surface area have been attributed to different applications, not all of them associated with the lattice coordination viewpoint. In the case of a molecule such as a polymer or a drug compound bathed in a solvent, the solvent accessible surface area is sometimes used. The solvent accessible surface area is based on the geometric interpretation of a solvent radius reducing the physical contact area between solvent and molecule. Thus, only at a solvent radius of zero is the solvent accessible surface area

equivalent to the exposed molecular surface area. In this work the simple exposed molecular surface area is used since it provides the clearest interpretation in a statistical mechanical sense.

The formal definition of the surface area of a component $c \in C_m$ in a molecule m is given by Eq. 19.

$$A_c \left[m : \{r_c, \mathbf{x}_c(m)\}_{c \in C_m} \right] = \int_{S_c^m} \min_{c^* \in C_m \setminus c} \{ \chi_{c,c^*}(\mathbf{x}) \} d\mathbf{x} \quad (19)$$

where

$$S_c^m \equiv \{ \mathbf{x} \mid \|\mathbf{x} - \mathbf{x}_c(m)\|_2 = r_c \}, \quad (20)$$

$$\chi_{c,c^*}(\mathbf{x}) = \begin{cases} 0 & \text{if } \|\mathbf{x} - \mathbf{x}_{c^*}(m)\|_2 < r_{c^*}, \text{ and} \\ 1 & \text{otherwise} \end{cases} \quad (21)$$

$$\|\mathbf{x}\|_2 \equiv \sqrt{\mathbf{x} \cdot \mathbf{x}} \quad (22)$$

A number of algorithms have been proposed for estimation of molecular surface area, including equi-spaced grids, Monte Carlo sampling, and semi-analytical methods. In this work a Monte Carlo strategy has been adopted for the numerical integration of Eq. 19. Monte Carlo integration involves the generation of a finite number of sample points, \mathbf{x}_i ($i = 1, \dots, SP$), on S_c^m through the use of a random number generator. The Monte Carlo approximation is depicted in Eq. 23.

$$A_c \left[m : \{r_c, \mathbf{x}_c(m)\}_{c \in C_m} \right] \approx \frac{1}{SP} \sum_{i=1}^{SP} \min_{c^* \in C_m \setminus c} \{ \chi_{c,c^*}(\mathbf{x}_i) \}, \quad \mathbf{x}_i \in S_c^m \quad (23)$$

This approach has an advantage over uniform discretization methods that new sample points can be added without regeneration of a mesh.

Generation of random variates on a spherical surface of a given component involves several steps:

- (i) generating Cartesian random variates $\langle x, y \rangle$ on a uniform $[0, 1] \times [0, 1]$ mesh;
- (ii) transforming the $\langle x, y \rangle$ to Cartesian points on the surface of a reference sphere of unit radius centered at the origin;
- (iii) translating of the sphere origin to the location of the component of interest; and
- (iv) scaling the sphere coordinates to the component radius.

The sampling strategy (step (i)) for generating random variates is considered below. Steps (iii) and (iv) are straightforward algebraic manipulations. Step (ii) is accomplished by Eqs. 24–27.

$$s_x = 2x - 1 \quad (24)$$

$$s_y = \sqrt{1 - s_x^2} \cos(\Omega) \quad (25)$$

$$s_z = \sqrt{1 - s_x^2} \sin(\Omega) \quad (26)$$

where

$$\Omega \equiv \pi(2y - 1) \quad (27)$$

The Monte Carlo strategy has several additional computational advantages over other methods to allow development of an efficient treatment of the $\min\{\chi\}$ terms. First, the order in which the $c^* \in C_m \setminus c$ are supplied to the indicator function $\chi_{c,c^*}(\mathbf{x}_i)$ can be heuristically adjusted to test for overlap with the closest neighboring components first. The heuristic consists of generating a preliminary set of sample points, \mathbf{x}_i ($i = 1, \dots, SP0$), for each $c \in C_m$ and counting the number of occurrences for which $\chi_{c,c^*}(\mathbf{x}_i) = 0$ for each $c^* \in C_m \setminus c$. Then the c^* are sorted in descending order of the number of occurrences for which $\chi_{c,c^*}(\mathbf{x}_i) = 0$, forming an ordered list of comparisons. A second computational advantage is that, unlike semi-analytical approaches, both the indicator function test and the integration over the individual atoms are fully parallelizable.

A Monte Carlo integration algorithm to evaluate Eq. 19 for a single molecule is given in Figure 5.4. The subscript notation associated with the molecule m is dropped to simplify the figure.

Figure 5.4. Program Design Language for Numerical Integration Algorithm.

```

for  $c \in C$  {
  for  $s = 1, \dots, SPO$  {
     $i \leftarrow 1$ ;
    while  $\left[ i < |C^*(c)| \text{ and } \chi_{c_i^*}(\mathbf{x}_c^s) = 0 \right]$  do {
       $i \leftarrow i + 1$ ;
       $hits_{c_i^*} \leftarrow hits_{c_i^*} + \chi_{c_i^*}(\mathbf{x}_c^s)$ ;
    }
  }
  sort  $\{i\}$  until  $hits_{c_i^*} \leq hits_{c_{i+1}^*}$ ,  $c_i^* \in C^*(c)$ ;
   $hits_c \leftarrow \sum_{i=1}^{|C^*(c)|} hits_{c_i^*}$ ;
  for  $s = 1, \dots, SPI$  {
     $i \leftarrow 1$ ;
    while  $\left[ i < |C^*(c)| \text{ and } \chi_{c_i^*}(\mathbf{x}_c^s) = 0 \right]$  do {
       $i \leftarrow i + 1$ ;
       $hits_c \leftarrow hits_c + \chi_{c_i^*}(\mathbf{x}_c^s)$ ;
    }
  }
   $sarea_c = \frac{hits_c}{SPO + SPI} \times 4\pi r_c^2$ ;
}

```

In this work a recently-introduced quasi-random Monte Carlo sampling strategy is employed. Specifically, shifted-Hammersley-Wozniakowski quasi-random numbers (HW; Wozniakowski, 1990) are used to generate sample points. Hammersley-Wozniakowski variates are known to have much reduced average-case complexity as compared to pseudo-

random number generators. To illustrate, a Monte Carlo algorithm employing pseudo-random generated variates has a complexity that grows exponentially with the dimensionality of the integral (Eq. 28).

$$SP \propto \Theta(\varepsilon^{-d}), \quad (28)$$

where

$$\varepsilon \propto \left| \frac{\int_{\text{True}} - \int_{\text{Monte Carlo}}}{\int_{\text{True}}} \right| \quad (29)$$

and d is the dimensionality of the integral. HW points, on the other hand, exhibit sub-exponential complexity in the average case (Eq. 30).

$$SP \propto \Theta\left(\varepsilon^{-1} \left\{ \ln \varepsilon^{-1} \right\}^{\frac{d-1}{2}}\right) \quad (30)$$

The reason for the improved performance of HW points is that the variates are more uniformly distributed than traditional pseudo-random variates. In fact, HW points can be considered intermediate between pseudo-random and uniformly generated points, as shown in Figure 5.5.

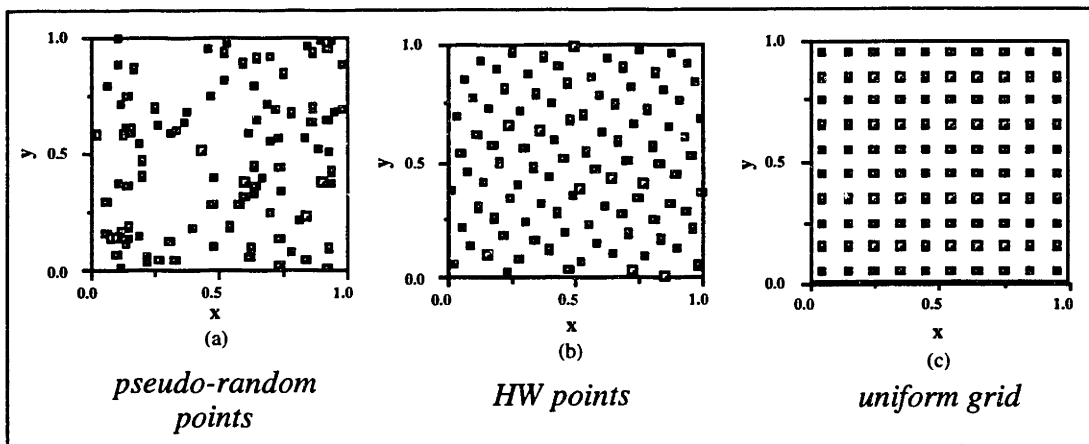


Figure 5.5. Comparison of Pseudo-Random (Multiplicative Generator) Points, HW Points, and a Uniform Grid.

Because of improved uniformity, HW points tend to give monotone error reduction in the integral as compared to pseudo-random numbers, in which the error norm tends to oscillate considerably and dampen out only after a large number of points are used. The HW points, by contrast, usually allow convergence in an essentially monotonic fashion, typically to 1-2% error within 200 sample points. It should be noted that shifted-

Hammersely points are optimized for determination of expected values. The sampling strategy does not seem to be a valid one for computation of higher moments such as the variance.

5.9. Optimization Algorithm

The general multivariate nonlinear regression problem to be solved is given in Eq. 31.

$$\min_{\{\alpha_n\}_{n \in N}, \{par_c\}_{c \in C}} \sum_{m=1}^{ND} [F_{exp}^m - F^m]^2 \quad (31)$$

where F_{exp}^m is an experimental measurement for the molecule m and F^m ($m = 1, \dots, ND$) is the predicted value as defined in Eq. 1. The optimization involves multiple local minima due to nonconvexity of F^m . In particular, there is a trilinear relationship established between α_n and par_c due to the form $\alpha_n \varphi_n[\bullet] \cdot \varphi_n[\bullet]$. Thus, if local nonlinear optimization methods are used then the fit obtained will generally require a systematic variation in the initial parameter guesses to find an acceptably robust choice of parameters. As described in the introduction, the parameters α_n are to be property-specific whereas the parameters par_c are hopefully transferable to multiple properties. The solution strategy adopted here is to solve for the α_n and par_c iteratively, allowing the user to specify an arbitrary subset of the par_c to have fixed parameters. In effect, Eq. 14 is reformulated as an iterative procedure between the optimization of the α_n and the par_c (Eq. 32).

$$\left. \begin{array}{l} \text{Linear Regression Step: } \min_{\{\alpha_n\}_{n \in N}} \{ \\ \text{Nonlinear Optimization Step: } \min_{\{par_c\}_{c \in C_A}} \sum_{i=1}^{ND} [F_{exp}^i - F^i]^2 \end{array} \right\} \quad (32)$$

The set C_A in Eq. 32 is the set of "Active" components whose parameters are to be optimized. The remaining parameters are fixed at their initial values. This approach facilitates model extension to additional atom-types and also facilitates adding or subtracting functions f_n to the model. Based on Eq. 32, statistical analysis is performed through linear regression in determining the α_n . Statistical results in the linear regression form a basis to compare alternate forms of the $\varphi_n[\bullet]$.

5.9.1. Levenberg-Marquardt Algorithm

Considered is the general nonlinear regression problem given in Eq. 33.

$$\min_z \sum_{i=1}^{NDAT} [g_{meas}^i - g^i(z)]^2, \quad (33)$$

A commonly employed method for solving Eq. 33 is the Levenberg-Marquardt method, which derives its search direction by combining the steepest descent and Newton directions. The gradient direction for Eq. 33 is given in Eq. 34.

$$\mathbf{f}(z) \equiv \sum_{i=1}^{NDAT} [g^i(z) - g_{meas}^i] \frac{\partial g^i(z)}{\partial \mathbf{z}} \quad (34)$$

For a given vector \mathbf{z} the steepest descent direction is $-\mathbf{f}(z)$. Newton's direction for minimization of Eq. 33 is given in Eqs. 35 and 36.

$$\Delta \mathbf{z} = -\mathbf{H}^{-1}(z) \mathbf{f}(z), \quad (35)$$

where

$$\mathbf{H}(z) = \mathbf{H}^T(z) = \frac{\partial \mathbf{f}(z)}{\partial \mathbf{z}^T} \quad (36)$$

is the Hessian of Eq. 33 (" T " denotes transposition of a vector or matrix). The Levenberg-Marquardt method starts with Newton's method and modifies the diagonal elements of the Hessian by a factor $(1 + \lambda)$. Denoting matrix elements of the Hessian by H_{ij} , the modification is given in Eqs. 37 and 38.

$$H_{ii} \leftarrow (1 + \lambda) H_{ii}, \quad i = 1, \dots, \dim(x) \quad (37)$$

$$H_{ij} \leftarrow H_{ij}, \quad i \neq j \quad (38)$$

The effect on the search direction and magnitude is given by Eqs. 39 and 40 (the use of a line search is not discussed here):

$$\lambda \rightarrow 0 : \Delta \mathbf{z} \rightarrow -\mathbf{H}^{-1}(z) \mathbf{f}(z) \quad (\text{Newton direction and step-size}) \quad (39)$$

$$\lambda \rightarrow \infty : \frac{\Delta \mathbf{z}}{\|\Delta \mathbf{z}\|_2} \rightarrow \mathbf{f}(z), \quad \|\Delta \mathbf{z}\|_2 \rightarrow 0 \quad (\text{Steepest descent direction}) \quad (40)$$

The Levenberg-Marquardt strategy allows for a particularly robust minimization algorithm. Initially, a search direction and magnitude close to the Newton direction can be tried by setting λ to a small value, say 0.001. If the resulting step does not lead to a decrease in Eq. 33, then that direction is not taken. Rather, the original Hessian is

modified by iteratively increasing λ , say by a factor of 10, until a decreasing search direction is found. If, on the other hand, the resulting step reduces Eq. 33, then that step is taken and the parameter λ increased by a factor of 10. By adjusting λ in this fashion, one is guaranteed to find a stationary point, *i.e.*, $f(\mathbf{z}) = 0$, which can either be a local minimum or a saddle point.

Assuming the measurement deviations are symmetrically distributed around zero, the form of Eq. 33 can be exploited to simplify the Hessian. Expanding the Hessian (Eq. 41),

$$\begin{aligned} H_{jk} &= \frac{\partial}{\partial z_j} \left\{ \sum_{i=1}^{NDAT} [g^i(\mathbf{z}) - g_{meas}^i] \frac{\partial g^i(\mathbf{z})}{\partial z_j} \right\} = \\ &= \sum_{i=1}^{NDAT} \left\{ \frac{\partial g^i(\mathbf{z})}{\partial z_j} \frac{\partial g^i(\mathbf{z})}{\partial z_k} + [g^i(\mathbf{z}) - g_{meas}^i] \frac{\partial^2 g^i(\mathbf{z})}{\partial z_j \partial z_k} \right\}, \end{aligned} \quad (41)$$

it is observed that the summation representing the second derivative term will tend to average out to zero, *i.e.*,

$$\sum_{i=1}^{NDAT} \left\{ [g^i(\mathbf{z}) - g_{meas}^i] \frac{\partial^2 g^i(\mathbf{z})}{\partial z_j \partial z_k} \right\} ca. 0. \quad (42)$$

Thus, a good approximation to the Hessian can be written as Eq. 43.

$$H_{jk} = \sum_{i=1}^{NDAT} \left\{ \frac{\partial g^i(\mathbf{x})}{\partial x_j} \frac{\partial g^i(\mathbf{x})}{\partial x_k} \right\} \quad (43)$$

By inspection of Eqs. 39 and 40, it is clear that the approximation (Eq. 43) does not change the stationary conditions for the solution. This approximation is important because it allows the Hessian to be calculated from the first derivatives and avoids the requirement of computing second derivatives. The calculation of second derivatives becomes computationally expensive in the case of large problems (*e.g.*, dozens of parameters).

5.9.2. Solution of the Outer Problem: Linear Regression

The outer problem consists of updating the α_n ($n = 1, \dots, N$) given the most recent numerical values of $f_n[m]$, denoted $f_n^*[m]$ ($m = 1, \dots, ND$). The problem is a linear regression, namely

$$\min_{\{\alpha_n\}_{n \in N}} \sum_{m=1}^{ND} \left[F_{exp}^m - \sum_{n=1}^N \alpha_n f_n^*[m] \right]^2. \quad (44)$$

5.10. Example: Estimation of Pure Component Properties

This section summarizes the results presented by Knight (1993) as an illustration of the model building strategy. A model form applicable to the estimation of both the critical temperature and the normal boiling point is presented. The critical temperature is modeled first to estimate parameters defining intermolecular interaction contributions. The transferability of the descriptors is then tested by their linear regression on the same set of contributions to predict the normal boiling point. The training set consists of 42 compounds, including 14 substituted aromatics, 9 hydrocarbons, 8 chlorohydrocarbons, 8 non-aromatic amines, and 3 cyano compounds (Table 5.2).

Table 5.2. List of Compound Classes and Compound Databank Entries Used in Model Development.

Compound Class	No. Compounds	Compound Databank Entries
hydrocarbons	9	21–29
chlorinated hydrocarbons	8	31–38
amines (non-aromatic)	8	41–48
nitriles	3	51–53
aromatics	14	1, 3, 5–7, 9–17

Model development is based on the linear property-energy relationship given in Eq. 45.

$$T_c \text{ or } NBP = \alpha_1 f_1(m) + \alpha_2 f_2(m) + \alpha_3 f_3(m) + \varepsilon, \quad (45)$$

where T_c denotes the critical temperature and NBP denotes the normal boiling point. Theoretical justifications for the functional form of Eq. 45 are given in the next chapter. The first function is a nonpolar dispersion interaction (Eq. 46).

$$f_1(m) = A(m) \quad (46)$$

The second function is a polar repulsive interaction energy contribution (Eq. 47),

$$f_2(m) = \sum_{c_1} \sum_{c_2} \left(b_2^{c_1} q_{c_1} p_{c_1} \right) \cdot \left(b_2^{c_2} q_{c_2} p_{c_2} \right), \quad (47)$$

$$c_1, c_2 \in C_m \text{ st. } \left(b_2^{c_1} q_{c_1} p_{c_1} \right) \cdot \left(b_2^{c_2} q_{c_2} p_{c_2} \right) > 0$$

and the third function is a polar attractive interaction energy contribution (Eq. 48).

$$f_3(m) = \sum_{c_1} \sum_{c_2} \left(b_2^{c_1} q_{c_1} p_{c_1} \right) \cdot \left(b_2^{c_2} q_{c_2} p_{c_2} \right), \quad (48)$$

$$c_1, c_2 \in C_m \text{ st. } \left(b_2^{c_1} q_{c_1} p_{c_1} \right) \cdot \left(b_2^{c_2} q_{c_2} p_{c_2} \right) < 0$$

The molecular parameters are the same types as those used by Grigoras (1990). Namely, $M_c^m = \langle A_c(m), q_c(m) \rangle$, where $q_c(m)$ is the partial atomic charge for $c \in C_m$, and $par_c = \langle b_c \rangle$, where b_c is the charge scaling factor. Molecular orbital calculations consist of semi-empirical AM1 calculations. The partial charge is determined by Mulliken population analysis. The atom-type radii are specified from correlation of the critical volume with the molecular surface area; the rationale for this strategy is given in chapter six. Optimized atom-type radii are given in Table 5.3.

Table 5.3. List of Input Model Parameters and van der Waals Radii.

Component	Optimized Radius (Å)	van der Waals Radius (Å)
<i>Hydrogen</i>		
aliphatic	1.6978	1.2
sp ²	1.8032	1.2
hydrogen bonded	1.3961	1.2
<i>Carbon</i>		
aliphatic	1.6930	1.7
aromatic	1.9411	1.7
sp	2.4347	1.7
<i>Nitrogen</i>		
sp ³	1.9767	1.55
sp ²	2.3360	1.55
sp	2.4116	1.55
<i>Oxygen</i>		
sp ²	2.1370	1.52
<i>Chlorine</i>		
sp ³	2.4139	1.75

Molecular orbital calculations are performed using GAUSSIAN 92 on a Cray C90 at the Pittsburgh Supercomputing Center. Model building computations are performed on a DEC ALPHA/AXP 4000. Typical computational times for a single molecular orbital

calculation are 30 CPU seconds on a single processor of the C90. A single evaluation of a molecular surface area is accomplished in under a tenth of a second on the ALPHA/AXP 4000 for the compounds considered. The complete model building exercise (*i.e.*, numerical integration, linear regression, and nonlinear regression) requires approximately five minutes and generally includes a dozen or more iterative optimization steps.

The best model obtained for the critical temperature and a comparison between the model and the method of Joback are given in Table 5.4. Figure 5.6 shows a comparison between estimated critical temperatures computed with the model and measured values for the set of 42 compounds.

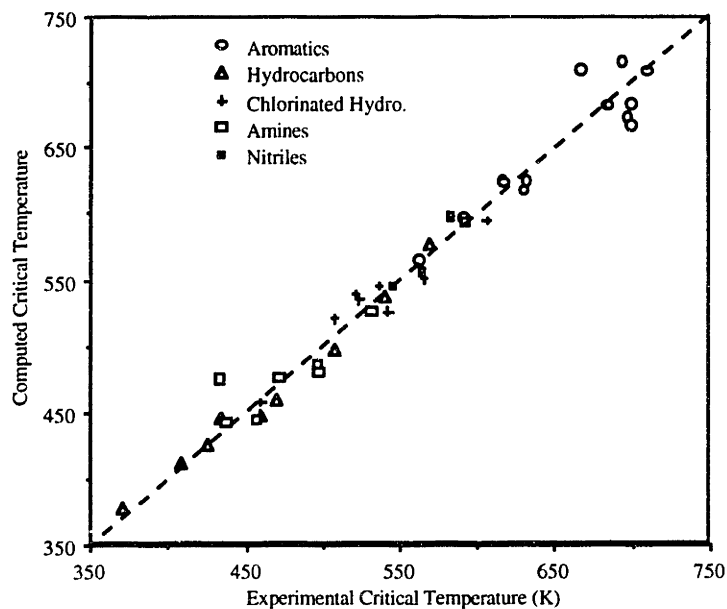


Figure 5.6. Comparison of Measured and Predicted Critical Temperatures for the Set of 42 Compounds.

It should be noted that unlike the methods of Ambrose and Joback (Reid *et al.*, 1988), the new model does not require the boiling point to compute the critical temperature. Furthermore, the total number of atom-types is smaller than the number of groups in group contribution methods.

The transfer of charge scaling factors to predict the normal boiling point is depicted in Figure 5.7. It is interesting to note that the model for the normal boiling point yields a smaller absolute error than for the critical temperature.

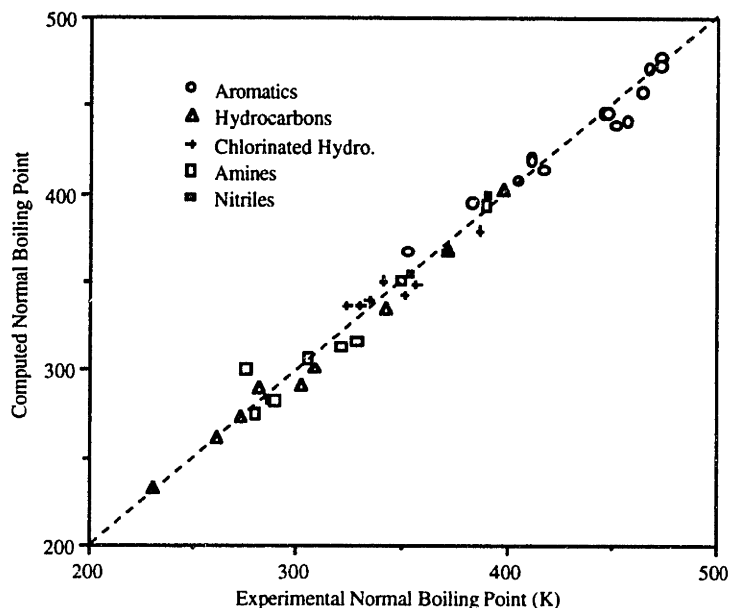


Figure 5.7. Comparison of Measured and Predicted Normal Boiling Points for the Set of 42 Compounds.

The final model for the normal boiling point, in which the charge scaling factors are optimized as part of the correlation procedure, is given in Table 5.4. The data used to develop the model consists of the original set of 42 compounds and 13 additional substituted anilines and nitrobenzenes. The charge scaling factors and descriptor values for the fits are presented by Knight (1993). It should be noted that a much larger data set (135 compounds), including hydrogen-bonding compounds, esters, *etc.* are treated in chapter six. In addition, chapter six considers extension of pure component property parameters to mixture properties such as partition coefficients and linear solvation-energy relationships.

Table 5.4. Summary of Results For Critical Volume, Critical Temperature, and Normal Boiling Point.

Description of Model	Mean Absolute Error, % (*)	Coefficient of Determination	Model Parameters			
			dispers.	polar attr.	polar repul.	const.
			$\hat{\alpha}_1$ (*)	$\hat{\alpha}_2$	$\hat{\alpha}_3$	$\hat{\epsilon}$
critical vol., cc mole ⁻¹	2.2 (2.6)	0.98	2.29 (0.05)	-	-	104 (9)
critical temperature, K	2.0 (4.0)	0.97	2.62 (0.09)	-93 (3)	-59 (3)	148 (15)
normal boiling point, K	1.9 (4.2**)	0.99	1.98 (0.05)	-55.0 (1.3)	-36.9 (1.5)	45 (8)

*Method of Joback.

**Normal boiling point used computed by method of Joback.

5.11. Discussion and Extensions

The computational framework provides the basis for systematic investigation of models incorporating parameters based on molecular calculations. Chapter six extends the results presented in this section to a much larger data set (135 compounds), including hydrogen-bonding compounds, esters, ethers, *etc.* Issues addressed in chapter six include:

- Testing the robustness of the model with compounds outside the "training set."
- Further analysis of the statistical significance of surface area and electron density measures in proposed models. It is of particular interest to determine whether terms of the form

$$p_{c_1}^m q_{c_1}^m b_{c_1} \cdot p_{c_2}^m q_{c_2}^m b_{c_2} \text{ or } A_c^m q_c^m b_c \quad (49)$$

are statistically superior to

$$p_{c_1}^m b_{c_1} \cdot p_{c_2}^m b_{c_2} \text{ or } A_c^m b_c. \quad (50)$$

- Comparison with group contribution methods containing a similar number of parameters.

5.12. Conclusions

A set of computational tools has been developed for construction and application of property models incorporating molecular calculations. A model structure has been chosen that includes explicit component-component interactions in the context of a statistical athermal coordination model. The multivariate nonlinear regression problem has been partitioned into a two-level problem involving (i) nonlinear regression of the component-component interaction parameters; and (ii) linear regression of apolar and polar-type contribution parameters. Performing regressions in this way allows comparison of different models through statistical analysis of the linear regression results. The Monte Carlo surface area integration strategy exploits a minimum average-case complexity sampling strategy using a presorting heuristic. Computational performance of the integration strategy shows that the method achieves 1-2% error usually within 200 sample points. Finally, the example problem has demonstrated that a simple model of intermolecular interactions can yield relatively accurate correlation of several pure component properties; thus the proposed class of models appears promising.

5.13. Supplement: Computer Implementation

5.13.1. Job Specification for Molecular Calculations

Molecular orbital calculations for sets of molecules are commonly required in both user-specified and application-specified work. A common interface to both types of work has been created to

- define a set of molecular structures and supply a guess at the approximate geometry;
- define component types;
- generate a set of input files to GAUSSIAN 92 and initiate a batch job; and
- control execution of the application as job results are retrieved and parsed.

An input file format has been created to allow the definition of functional groups, component types, and molecular structures. This file builds the molecular structures from two types of structures, <fgroup> and <molecule>, constructed from six types of objects. The objects are described in Table 5.5.

Table 5.5. Objects Used in Job Definition.

<fname>	(fragment name);
<a#>	(atom type; # denotes an integer index);
<cid#>	(component identifier);
<fcoord#>	(reference Cartesian coordinates of an atom);
<mname>	(molecule name); and
<fplace#>	(orientation of a fragment in a molecule).

The <fgroup> structure defines molecular fragments (or an entire molecule) in terms of atom-types and component specifiers. The general form of this structure is as follows:

Table 5.6. Definition of <fgroup> Structure.

fgroup <i>fname</i>	<i>a1</i> <i>cid1</i>	<i>a2</i> <i>cid2</i>	...	<i>an</i> <i>cidn</i>	x 1
	<i>x1</i>	<i>y1</i>	<i>z1</i>		
	<i>x2</i>	<i>y2</i>	<i>z2</i>		
		...			
	<i>xn</i>	<i>yn</i>	<i>zn</i>		

The term *fname* is the fragment identification name. The objects *a1*, *a2*, etc. are atom-types and are required to conform to acceptable GAUSSIAN input format (e.g., c, C, Cl, CL). The objects *cid1*, *cid2*, etc. are integer-valued component specifier indices defined for each atom-type. It is these indices that provide a mapping of computational results from the

molecular calculations onto the M_c^m vector of component attributes within molecules. The *cid* values are built into the coordinate variable declarations in GAUSSIAN input files and are parsed directly from the output files. Currently up to eight different component types may be defined for each atom M_c^m .

The first line is terminated by "x #"; (the number, #, is arbitrary but must be integer-valued). The remaining lines of the <fgroup> structure define the geometry of the fragment in reference cartesian coordinates measured in Ångstroms. Generally, the appending atom on the fragment is located at the origin. And the remaining atoms oriented in an obvious fashion that seems best suited for attachment to other fragments. Two typical examples are given in Figure 5.8.

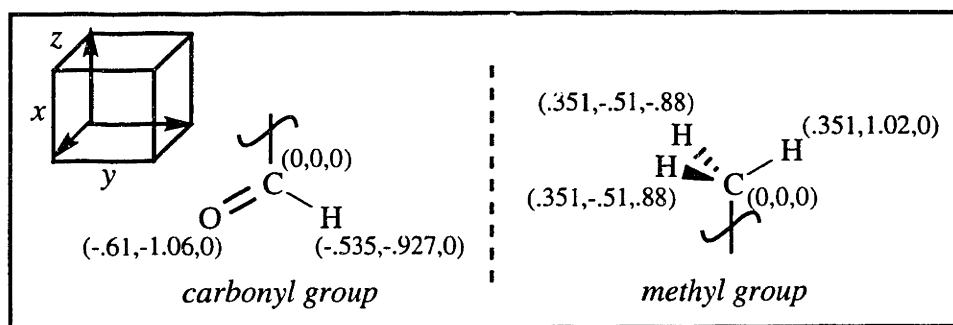


Figure 5.8. Examples of Fragments in Their Reference Cartesian Coordinates (Å).

A number of fragment definitions are currently available in several job files. The cartesian coordinates are either based on ideal bond angles and lengths or on AM1 semi-empirical optimized fragments taken from reference molecules. A list of available groups is given in Figure 5.9.

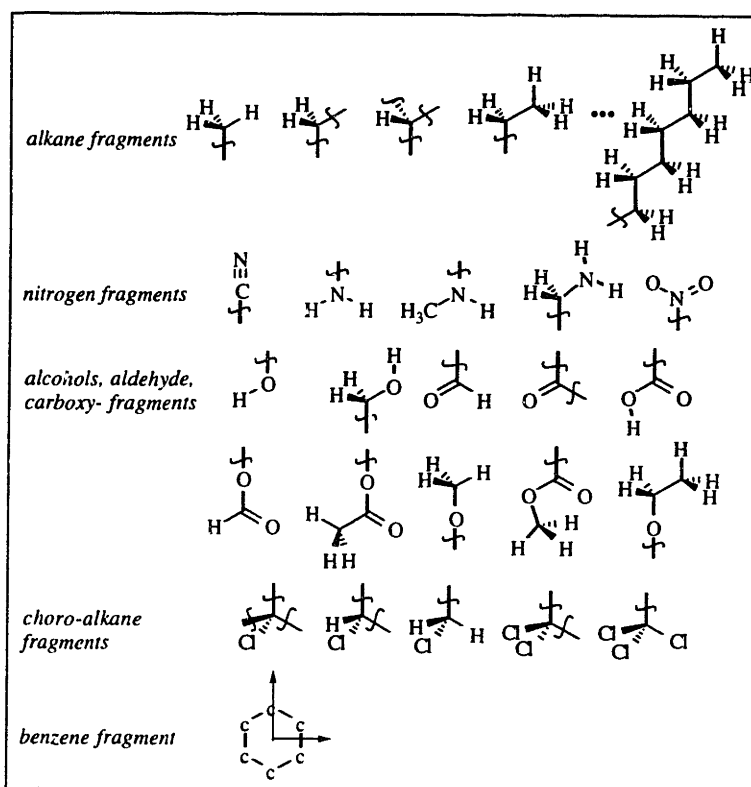


Figure 5.9. Sample Fragments Available in Job Specification Files.

The <molecule> structure assembles a set of fragments for form a molecule. The general form of this structure is as follows:

Table 5.7. Definition of <molecule> Structure.

molecule	mname	fname1	fname2	...	fnamem	x
raz1	rax1	ray1	xt1	rbz1	rbx1	rby1
raz2	rax2	ray2	xt2	rbz2	rbx2	rby2
			...			
razm	raxm	raym	xtm	rbzm	rbxm	rbym

The <molecule> structure consists of a list of fragments terminated by "x". Each fragment is oriented from its reference coordinates into position by a series of rotations and translation. The sequence of operations is depicted in Figure 5.10.

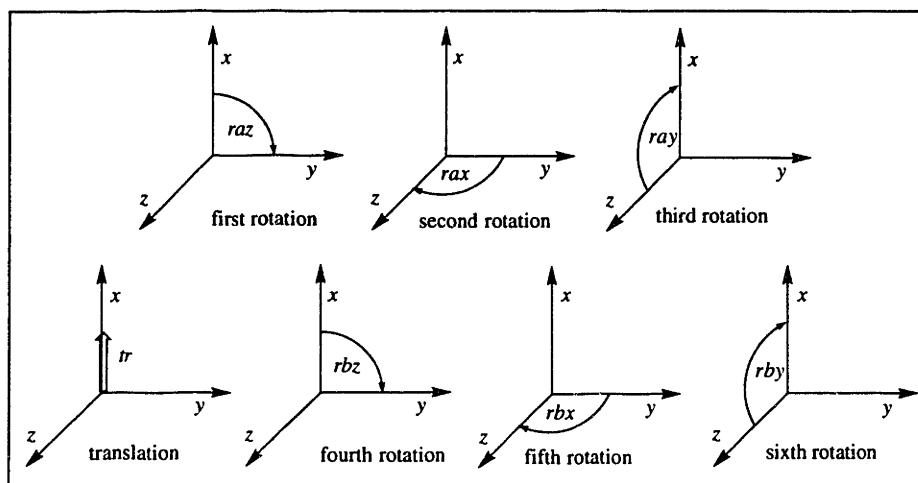


Figure 5.10. Sequence of Operations That Can Be Performed on Each Fragment.

In applications jobs, cartesian coordinates for the set of molecules is often available. Thus, molecule specification consists of defining the molecule as a fragment followed by the following trivial definition.

Table 5.8. Specification of a Molecule Fragment.

molecule	mname	molecule_fragment	x	
0.	0.	0.	0.	0.

An example of job specification for three molecular fragments and two aldehydes is given in Table 5.9.

Table 5.9. Example of Job Specification File: A Couple of Aldehydes.

fgroup co	c 2	o 2	x 1				
0.	0.	0.	0.				
-0.61	-1.0566	0.					
fgroup cho	c 2	o 2	h 2	x 1			
0.	0.	0.	0.				
-0.61	-1.0566	0.					
-0.535	0.9266	0.					
fgroup ch3	c 1	h 1	h 1	h 1	x 1		
0.	0.	0.	0.				
0.3516	1.0212	0.					
0.3516	-0.5106	0.8843					
0.3516	-0.5106	-0.8843					
molecule ald1	cho	ch3	x				
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	1.51	0.	0.	0.	0.
molecule ald2	co	ch3	ch3	x			
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	1.51	0.	0.	0.	0.
0.	0.	0.	1.51	120.	0.	0.	0.

The GAUSSIAN 92 input file generated for acetone (CO(CH₃)₂) is shown in Table 5.10.

Table 5.10. GAUSSIAN 92 Input File for AM1 Geometry Optimization in Cartesian Coordinates.

```
# AM1 Opt=(Cartesian,NOEIGENTEST)

COMPUTATION SPAWNED FROM RETROSYNTHETIC GENERATOR

0, 1
c 0 t2x1 y1 z1
o 0 t2x2 y2 z2
c 0 t1x3 y3 z3
h 0 t1x4 y4 z4
h 0 t1x5 y5 z5
h 0 t1x6 y6 z6
c 0 t1x7 y7 z7
h 0 t1x8 y8 z8
h 0 t1x9 y9 z9
h 0 t1x10 y10 z10

t2x1 0.000000
y1 -0.000200
z1 0.000000
t2x2 -0.610000
y2 -1.056600
z2 0.000106
...
t1x10 -0.488638
y10 1.867495
z10 -0.884348
```

5.13.2. Specification of Model Parameters

Specification of $\{par_c\}_{c \in C}$ is accomplished through an ascii input file. There is one main structure, <cname>, consisting of a list of parameter objects, par_j , $j = 1, \dots, s$. The general form of an entry in the input file is given in Table 5.11.

Table 5.11. Format of Component Parameters Specification File.

<i>cname</i>	<i>par</i> ₁
	<i>par</i> ₂
	...
	<i>par</i> _s

The set of parameters that are optimized and the set that are held fixed depends on the job. For those that are adjusted, the input values are taken as initial guesses. For example, if the framework were used for Grigoras' model, the atomic radii would be taken as fixed and the input charge scaling factors would be treated as initial guesses. It should also be noted that a subset of parameters in the file might not be used at all, depending on the definitions of the property functions.

*Table 5.12. An Excerpt From a Component Parameters Specification File.
For each component, the first number is the component radius; the second
number is the charge scaling factor.*

hali	1.6978
	1.393178
hsp2	1.795
	1.652953
hhb	1.839727
	2.313282
...	...
nsp3	2.079
	-2.819382
nar	2.336
	-1.760019
nsp	2.479
	-0.858291
nxx	0.
	0.

Specification of the dataset consists of an ascii file containing references to appropriate GAUSSIAN 92 output files. Fitting data pertaining to the dataset is also supplied in an ascii file.

Chapter 6

Integrated Model Building Strategy Incorporating Molecular Calculations. Theory, Results, and Extensions.

6.1. Introduction

Based on the computational tools developed in chapters four and five, this chapter presents computational experience with a modeling strategy incorporating molecular quantum parameters for a class of thermodynamic models. Property models incorporating molecular calculations currently include quantitative structure-property or activity relationships, multivariate regression of empirical molecular descriptors (Grigoras, 1990; Pulitzer, 1993), and principle component analysis (Grigoras, 1985). With the growing interest in such models, there is a critical need to address practical "model-building" issues associated with quantum calculations in terms of the scope and limitations of available methods. Vacuum molecular orbital calculations, for example, do not necessarily come close to the average solvated electron density nor the average structural conformation. As a result, there is a need to assess the statistical significance of the electronic parameters in models, preferably including a comparison to empirically fit parameters in similarly structured models. Furthermore, both semi-empirical and *ab-initio* methods exhibit computational artifacts that are specific to the choice or parameterization of the basis set. Parameter fits in property models tend to be specific to these artifacts. An additional criteria for judging proposed models is the capability for continual improvement and extension other properties with similar thermodynamic dependencies.

The strategy presented in chapter five exploits the interaction potential as a basis for development of improved thermodynamic models incorporating quantum calculations. Many such problems are ideally characterized as the reduction of complex general models to simpler ones incorporating measurement- and computation-derived parameters. The reduction is accomplished by taking advantage of a combination of simplifying attributes of compound classes and modeling assumptions. The resulting model forms are parameterized through appropriate reference states and model compound data. A depiction of the main steps in model development are depicted in Figure 6.1.

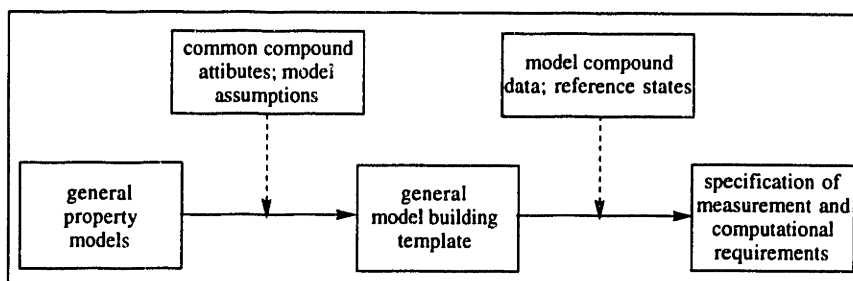


Figure 6.1. Steps in Model Development Incorporating Both Theory and Experiment.

An advantage of the model building strategy in Figure 6.1 is that quantum-derived parameters can be compared with purely empirical fits in property models. Furthermore, it is potentially possible to exploit theoretical relationships between thermodynamic properties to isolate a common set of parameters. Application to several properties should provide a basis for choice of model forms, computational methods, and parameterizations.

This chapter proposes a model of intermolecular interactions to be incorporated in the UNIQUAC lattice potential. The overall modeling strategy is depicted in Figure 6.2.

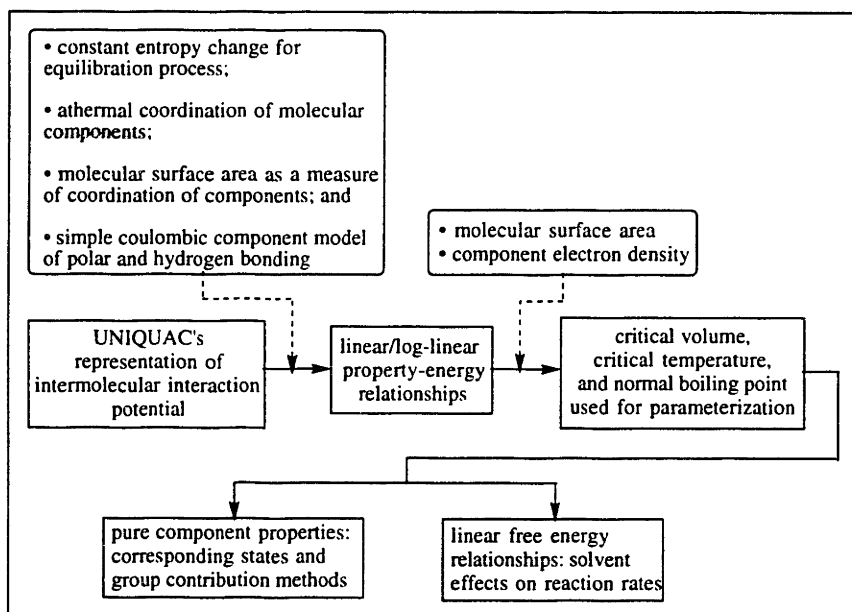


Figure 6.2. Property Estimation Strategy for Incorporation of Quantum-Derived Parameters.

The proposed model approximates dispersion, polar, and hydrogen-bonding interactions in a single lattice cell through defined component-component interactions. The model is intended for molecules above a minimum molecular weight, say 50, so that local component interactions rather than induction and orientation forces dominate the potential

energy (Israelachvili, 1992). Several assumptions are used to simplify the property model formulation, including constant entropy of vaporization (*i.e.*, Trouton's rule) and an athermal approximation of the local composition fractions for component-component coordination. The result of the assumptions is a class of linear property-energy relationships, among them a new model for the normal boiling point. Such relationships facilitate parameterization of the intermolecular potential model, which can, in principle, be incorporated in more complex property models. As an extension, the interaction parameters are incorporated into a model for solvent effects on reaction rates (a linear solvation-energy relationship) and for prediction of the octanol-water partition coefficient for a class of nitroaromatics.

The organization of the chapter is as follows. Linear relationships for the critical temperature and normal boiling point are first reviewed. Next, the proposed modeling formalism is presented and demonstrated for a set of 135 compounds. The statistical significance of the electron density parameters is demonstrated. Finally, theory and computational results are presented for the prediction of partition coefficients and estimation of solvent effects on reaction rates.

6.2. Overview

A new class of thermodynamic models incorporating molecular calculations is proposed for use in an industrial research and development setting. The components of the strategy are (i) use of relationships between pure component properties and the intermolecular potential energy to derive empirical energy parameters; (ii) use of molecular orbital calculations to obtain molecular surface area and electron density parameters; and (iii) extension of the interaction parameters to predict additional properties. The sensitivity of the electron density to the overall molecular structure accounts for variations in correlated properties that are averaged out by traditional group contribution methods. The statistical significance of geometric parameters and the electron density has been demonstrated for a simple constitutive model based on an analogy to the coulombic interaction.

A data set of approximately one hundred and forty compounds, including aliphatic hydrocarbons, olefins, chlorinated compounds, amines, aldehydes, alcohols, carboxylic acids, and polysubstituted aromatics has been constructed. The data includes a molecular weight distribution *ca.* 50–200, boiling point range *ca.* 230–570 K, and a molecular surface area range *ca.* 120–300 Å². The coefficient of determination for the model without alcohols and carboxylic acids is >0.99 (chapter five). With alcohols and carboxylic acids a coefficient of 0.98 is observed. Comparisons with Joback's group contribution are performed along with other tests to form an independent evaluation of the model.

Accurate correlation of the normal boiling point allows intrinsic interaction energies to be extracted for development of quantitative property and reactivity models. Having shown the transferability to critical properties in chapter five, transferability to

several mixture properties is demonstrated in this chapter. Of particular interest has been success in using a self-similar solvent measure in a linear solvation-energy relationship for the Menshutkin reaction. Also considered has been the prediction of octanol-water partition coefficients for a class of polysubstituted nitroaromatics. In both cases the proposed models compare well with previous approaches.

6.3. Role of Interaction Potential in Property Relationships

An approximate linear relationship between the normal boiling point and the interaction potential is readily established for relatively rigid, small organic compounds. The criteria for vapor-liquid equilibrium in a single component system is given in Eq 1.

$$\Delta G = \Delta H - T\Delta S = 0 \quad (1)$$

Solving for the temperature yields Eq. 2.

$$T = \frac{\Delta H}{\Delta S}, \quad (2)$$

where $\Delta H = \Delta U + P\Delta V$. Both the magnitude and variance of $P\Delta V$ tend to be much smaller than ΔU . Hence, to a first approximation,

$$T = \frac{\Delta U + \varepsilon}{\Delta S}, \quad (3)$$

where the small parameter ε represents an averaged $P\Delta V$ effect.

At a particular temperature, in this case the normal boiling point, the average kinetic energies of the two phases are equal and must cancel, as the internal molecular energies are assumed to do. Thus

$$\Delta U = \mu_l^0 - \mu_v^0, \quad (4)$$

where μ_l^0 and μ_v^0 are the intermolecular potential energies of interaction per molecule. Under conditions well removed from the critical state and assuming the vapor phase behaves as an ideal gas, μ_v^0 becomes zero. The resulting equation for the boiling point is

$$T = \frac{\mu_l^0 + \varepsilon}{\Delta S} \quad (5)$$

Since the liquid phase is always denser than the vapor phase, the denominator of Eq. 5 is negative in sign. Recognizing that for a liquid phase to be present $\mu_l^0 < 0$, it follows that $T > 0$. It is well known that for a variety of organic liquids the entropy of vaporization is approximately constant (*i.e.*, Trouton's rule; Reid *et al.*, 1987). Thus, Eq. 5 can be rewritten approximately as Eq. 6.

$$T = |\mu_l^0| \cdot \text{constant}_1 + \text{constant}_2 \quad (6)$$

A similar linear relationship between the critical temperature and the interaction potential can be established at a variety of levels of theoretical rigor. Statistical mechanics-derived models are available for lattice gases (Rice, 1967). For present purposes, the most straightforward illustration is the linear relationship between the critical temperature and the constants of the van der Waals equation (Eq. 7).

$$T_c = 8a/27Rb, \quad (7)$$

where the parameter a is an energy density, with units Energy – volume/mole², and b is the hypothetical volume occupied by one mole of molecules with no free space.

Assuming the molar critical volume is proportional to b ,

$$\mu_c^0 \propto \frac{a}{b}. \quad (8)$$

The assumption has a significant physical interpretation. A distinguishing feature between the interaction energies in the critical state and sub-critical equilibrium states such as the normal boiling point is the larger fraction of voids surrounding each molecule in the critical state. In statistical lattice theory, the void fraction effects the number of intermolecular contacts and, by implication, the combinatorial entropy. In such a model, the assumption that the molar critical volume is proportional to b is tantamount to assuming that the fraction of lattice vacancies, $n_0 = N_0/N$, partitioned into volumes equal to the volume of the molecule, is invariant for different molecules. This result implies Eq. 9.

$$\mu_c^0 \propto \mu_l^0 \quad (9)$$

One of the demonstrations in chapter five has been to show that parameters computed for one of the temperatures are indeed transferable to the other.

6.4. Characterization of Intermolecular Potential

The proposed empirical interaction energy model is based on the consideration of steric, dispersion, electrostatic, and hydrogen-bonding interactions. Short range repulsive interactions are characterized by assigning each molecular component an effective radius, which, along with the molecular configuration determines its exposed surface area. Due to vacancies present in the system, the predicted radius of an atom in the condensed state (under the artificial assumption of no vacancies) can be defined in a way that differs somewhat from the van der Waals radius, which is derived from the solid state. In particular, a reference thermodynamic state can be chosen (*e.g.*, the critical state) and the radius adjusted at that state to eliminate the need to explicitly account for the vacancy distribution (although this approach creates difficulties in theoretical derivations). As might be expected, the empirically determined radii are generally larger than van der Waals radii. A notable exception is hydrogen-bonding, where intermolecular distances can be smaller than the sum of van der Waals radii. It is noted that at molecular weights below about 50, orientational forces due to the molecular dipole and induced by the surrounding medium tend to be major contributors to the free energy in polar molecules; such forces are not explicitly handled in the proposed lattice representation of the intermolecular potential.

The definition of atomic radii for different hybridization states has been treated by Grigoras (1990) through the correlation of molecular surface areas with the critical volume. The linear relationship is supported by van der Waals equation, which predicts that the critical volume is proportional to the molecular volume, b (Eq. 10).

$$V_c = 3b \quad (10)$$

If the molecule to which Eq. 10 refers is treated as a polysegmented molecule distributed on a lattice, then linear relationships presented by Staverman (1950) establish proportionality between b and the exposed molecular surface area.

Recall from chapter five that the surface area, $A(m)$, of a molecule m consisting of components $c \in C_m$ in a reference geometry is given in Eq. 11.

$$A \left[m : \{r_c, \mathbf{x}_c(m)\}_{c \in C_m} \right], \quad (11)$$

where r_c is the radius and $\mathbf{x}_c(m)$ is the vector defining the coordinate of the component $c \in C_m$ in the reference geometry. The optimization of the radii to reproduce critical volumes is generally a reasonable approach, although significant deviations are observed for systems with strong hydrogen-bonding or charge-transfer effects. Phenol, for example, has a larger molecular surface area than benzene, but a substantially lower critical volume.

$$A(\text{phenol}) = 162.6 \text{ \AA}^2 ; V_c(\text{phenol}) = 229.0 \text{ cc mol}^{-1}$$

$$A(\text{benzene}) = 158.6 \text{ \AA}^2 ; V_c(\text{benzene}) = 259.0 \text{ cc mol}^{-1}$$

Results obtained for a set of non-oxygen-containing compounds are given in Figure 6.3. The forty-two compounds used in the fit are taken from the compound databank (appendix one) and include compounds 1, 3, 5–7, 9–17, 21–29, 31–38, 41–48, and 51–53.

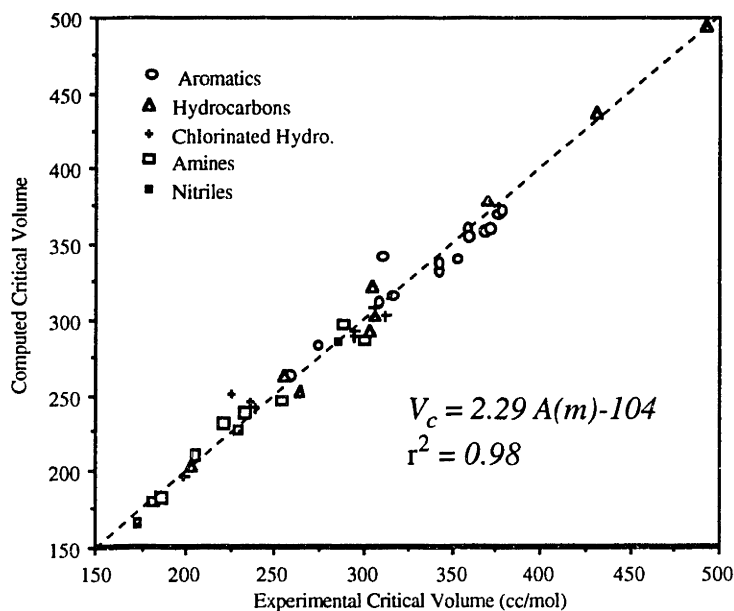


Figure 6.3. Correlation of Critical Volume as a Linear Function of the Computed Surface Area for Forty-Two Compounds.

The geometric structures for the molecules have been determined by AM1 geometry optimizations.

The optimized radii used in this work are a combination of our own calculations and those due to Grigoras (1990). The values used are reported in Table 6.1.

Table 6.1. Comparison of Computed Radii and van der Waals Radii.

Type	Optimized Radius (Å)	van der Waals Radius (Å)
<i>Hydrogen</i>		
aliphatic	1.6978	1.2
sp ²	1.8032	1.2
hydrogen-bonded	1.3961	1.2
<i>Carbon</i>		
aliphatic	1.6930	1.7
aromatic	1.9411	1.7
sp ²	2.0200	1.7
sp	2.4980	1.7
<i>Nitrogen</i>		
sp ³	1.9767	1.55
sp ²	2.3360	1.55
sp	2.4120	1.55
<i>Oxygen</i>		
sp ³	1.6720	1.52
sp ²	2.1370	1.52
hydrogen-bonded	1.6720	1.52
<i>Chlorine</i>		
sp ³	2.4140	1.75

By establishing effective atomic radii, one is not explicitly predicting short-range forces, but is rather specifying the degree to which groups can coordinate in the liquid phase. As shown in chapter five, the degree of coordination is essential in the evaluation of the lattice potential. The ability to define atomic contributions to the overall coordination allows the molecular components $c \in C_m$ in a molecule m to be defined as the atoms themselves (in a particular hybridization or bonding state). The resulting interaction between two components, $U_{c_1 c_2}(m)$, in a molecule m is given by Eq. 12.

$$U_{c_1 c_2} = z A(m) u_{c_1 c_2} \theta_{c_1 c_2}(m) \quad (12)$$

where z is the coordination number of the lattice, $A(m)$ is the molecular surface area, $u_{c_1 c_2}$ is the intrinsic interaction energy, and $\theta_{c_1 c_2}(m)$ is the component fraction between components c_1 and c_2 . Thus, the modeling problem reduces to estimation of $u_{c_1 c_2}$.

The term $u_{c_1c_2}$ is treated as a symmetric operator (i.e., $u_{c_1c_2} = u_{c_2c_1}$) consisting of four contributions:

- a nonpolar dispersion contribution, $u_{c_1c_2}^1$;
- a polar repulsive contribution, $u_{c_1c_2}^2$;
- a polar attractive contribution, $u_{c_1c_2}^3$; and
- a hydrogen-bonding contribution, $u_{c_1c_2}^4$.

In order to establish the statistical significance of quantum-derived parameters, careful attention should be given to minimizing the number of adjustable parameters. The basic strategy used has been to separate the binary interaction terms into a product of single component terms.

The nonpolar dispersion contribution is treated as a uniform, constant affect on each component. Thus,

$$u_{c_1c_2}^1 = 1. \quad (13)$$

The polar repulsive term, $u_{c_1c_2}^2$, is modeled by analogy with a repulsive coulombic interaction. Namely, the interaction between two components is proportional to the product of the partial charges (Eq. 14).

$$\begin{aligned} u_{c_1c_2}^2(m) &= (b_{c_1} q_{c_1}^m) \cdot (b_{c_2} q_{c_2}^m), \\ c_1, c_2 \in C_m \text{ st. } &(b_{c_1} q_{c_1}^m) \cdot (b_{c_2} q_{c_2}^m) > 0 \end{aligned} \quad (14)$$

where q_c^m is the partial charge and b_c^c the charge scaling factor of component $c \in C_m$. Similarly, the polar attraction term, $u_{c_1c_2}^3$, takes the form given in Eq. 15.

$$\begin{aligned} u_{c_1c_2}^3(m) &= (b_{c_1} q_{c_1}^m) \cdot (b_{c_2} q_{c_2}^m), \\ c_1, c_2 \in C_m \text{ st. } &(b_{c_1} q_{c_1}^m) \cdot (b_{c_2} q_{c_2}^m) < 0 \end{aligned} \quad (15)$$

Finally, the hydrogen-bonding term takes the same form as Eqs. 14 and 15 (a proposed strategy to define the hydrogen-bonding and charge-transfer contributions in terms of isodesmic formal reactions is discussed in the future work section) but is defined

only for specific O...H interactions. Specific binary interaction parameters are defined for these interactions.

$$u_{c_1 c_2}^4(m) = B_{c_1 c_2} q_{c_1}^m q_{c_2}^m, \quad (16)$$

$c_1, c_2 \in C_m$ st. c_1, c_2 form an H-bonding O...H pair

where the $B_{c_1 c_2}$ are binary interaction parameters. It should be noted that a special hydrogen-bonding parameter was also introduced for the N...H interaction but was subsequently shown not to be statistically significant. A variation in which the product is divided by the sum of the two component's radii was also tested, but was found to neither hurt nor significantly help the fit obtained.

Inserting Eqs. 13–16 into the expression for the lattice potential (Eq. 12), four distinct descriptors are generated (Eqs. 17–20).

$$f_1(m) = A(m) \quad (17)$$

$$f_2(m) = \sum_{c_1} \sum_{c_2} (b_{c_1} q_{c_1}^m p_{c_1}^m) \cdot (b_{c_2} q_{c_2}^m p_{c_2}^m), \quad (18)$$

$c_1, c_2 \in C_m$ st. $(b_{c_1} q_{c_1}^m p_{c_1}^m) \cdot (b_{c_2} q_{c_2}^m p_{c_2}^m) > 0$

$$f_3(m) = \sum_{c_1} \sum_{c_2} (b_{c_1} q_{c_1}^m p_{c_1}^m) \cdot (b_{c_2} q_{c_2}^m p_{c_2}^m), \quad (19)$$

$c_1, c_2 \in C_m$ st. $(b_{c_1} q_{c_1}^m p_{c_1}^m) \cdot (b_{c_2} q_{c_2}^m p_{c_2}^m) < 0$

$$f_4(m) = \sum_{c_1} \sum_{c_2} B_{c_1 c_2} (q_{c_1}^m p_{c_1}^m) \cdot (q_{c_2}^m p_{c_2}^m), \quad (20)$$

$c_1, c_2 \in C_m$ st. c_1, c_2 form an H-bonding O...H pair

The individual contributions defined by Eqs. 17–20 are related to properties of interest through a linear fit (e.g., Eq. 21).

$$T_c \text{ or } NBP = \alpha_1 f_1(m) + \alpha_2 f_2(m) + \alpha_3 f_3(m) + \alpha_4 f_4(m) + \varepsilon, \quad (21)$$

where T_c denotes the critical temperature and NBP denotes the normal boiling point. Of particular interest in the analysis of Eq. 21 is to determine the statistical significance of the

parameters $\alpha_1 - \alpha_4$ and ϵ . Furthermore, it is of interest to determine the effect of removing the inverse radius dependence on the fit, as well as removing the partial component charges.

Limitations present in the above model should be pointed out. Equation 16 is generally used for nearest neighbor interactions between different molecules. In other words, only interactions in the first solvent shell are treated. The functional form does not necessarily account for fixed dipolar interactions in the individual group contributions. Additionally, the athermal assumption, which seems appropriate most of the time for critical properties, is not the best treatment of the combinatorial entropy in normal condensed systems of polar molecules. These limitations suggest in particular that the model has an inherent molecular weight cutoff and should be less accurate for molecules that undergo dimerization. For example, accurate prediction of the properties of water would not be expected.

6.5. Pure Component Property Predictions

The proposed model has been tested for a diverse set of 135 organic compounds with normal boiling points in the range *ca.* 200–600K and molecular weights *ca.* 50–200. Compound classes considered are listed in Table 6.2.

Table 6.2. List of Compound Classes and Compound Databank Entries Used in Model Development.

Compound Class	No. Compounds	Compound Databank Entries
hydrocarbons	23	1, 7, 9–11, 21–29, 142–147, 159–161
chlorinated hydrocarbons	12	6, 15–17, 31–38
amines	17	3, 12–14, 41–48, 54–56, 64–65, 69
nitriles	6	5, 51–53, 63, 69
nitro-compounds	7	2, 60–62, 66–68
OCO-containing*	28	71–98
alcohols	26	4, 18–20, 101–114, 116–123
aldehydes	8	131–137, 139
ethers	8	151–158

*carboxylic acids, esters, *etc.*

Among the compound classes absent from the dataset are fluorinated compounds and those containing second row atoms (with the exception of chlorine).

Initial results are compared to Joback's group contribution method. Accuracies of the results are also evaluated for specific compound classes (*e.g.*, aldehydes and ethers) as a way to determine model applicability to different dominant interactions. In order to determine the robustness of the fit, a "training set" consisting of 95 compounds is constructed and the resulting model tested on 40 compounds outside the training set.

Finally, it is of interest to characterize the statistical significance of the electron density parameters. A reference model of identical structure to the proposed model but with partial charges set to unity is fitted to the data for comparison.

Molecular orbital calculations are performed in GAUSSIAN 92 on the Cray C90 at Pittsburgh Supercomputing Center. Model building computations are performed on a DEC ALPHA/AXP 4000.

6.5.1. Model Results

An initial fit of the dataset to the proposed model (Eq. 21) yields a mean absolute error of 9.1K (ca. 2.4%) and a coefficient of determination (r^2) of 0.977. Table 6.3 summarizes the model parameters obtained.

Table 6.3. Model Parameters and Statistics Obtained for the Dataset.

Mean Absolute Error (K)	Coefficient of Determination	Model Parameters				
		dispers.	polar attr.	polar repul.	hyd.-bonding	const.
		$\hat{\alpha}_1$ (*)	$\hat{\alpha}_2$	$\hat{\alpha}_3$	$\hat{\alpha}_4$	$\hat{\epsilon}$
9.1 (2.4%)	0.977	1.73 (0.04)	-47.2 (.1)	-15.8 (.6)	-31 (2)	101 (6)

*Standard Error.

A plot of the model predictions as compared to measured values is given in Figure 6.4.

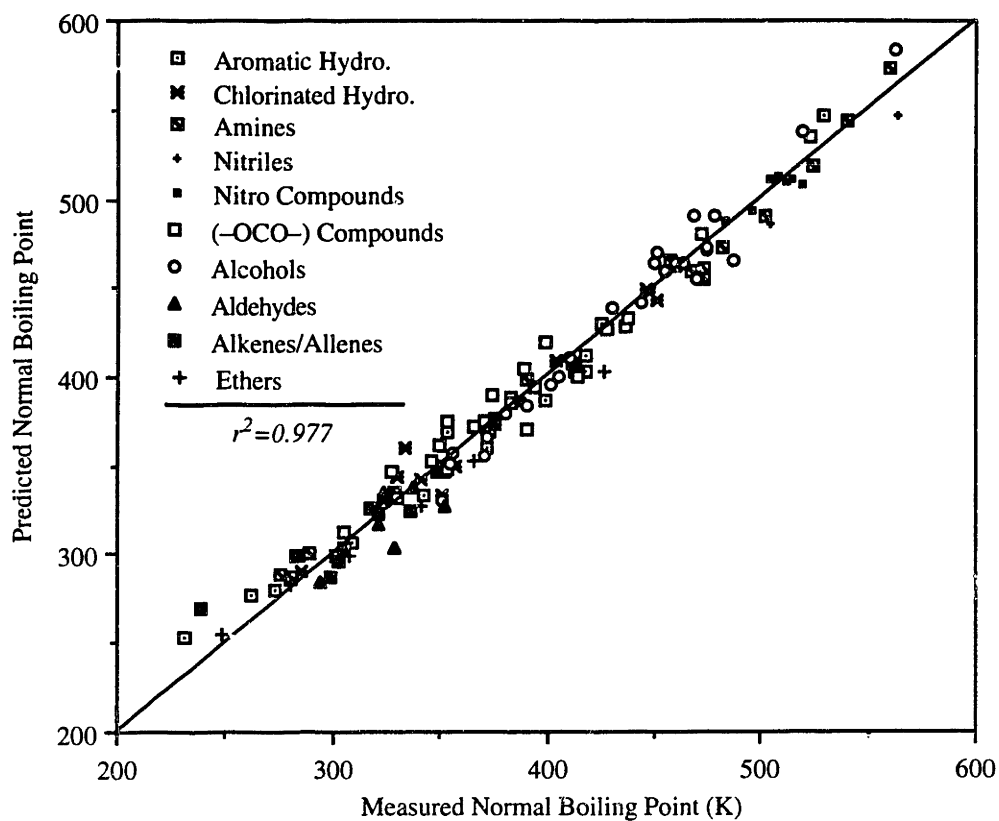


Figure 6.4. Fit of the Normal Boiling Point to the General Model (Eq. 21) for 135 Compounds.

The optimized charge scaling factors are given in Table 6.4.

Table 6.4. List of Input Radii and Computed Charge Scaling Factors.

Component	Optimized Radius (Å)	Charge Scaling Factor
<i>Hydrogen</i>		
aliphatic	1.6978	1.641
sp ²	1.8032	1.737
hydrogen-bonded	1.3961	-1.123
<i>Carbon</i>		
aliphatic	1.6930	1.000
aromatic	1.9411	1.000
sp ²	2.0200	0.266
sp	2.4980	2.498
<i>Nitrogen</i>		
sp ³	1.9767	-1.090
sp ²	2.3360	-0.105
sp	2.4120	-0.703
<i>Oxygen</i>		
sp ³	1.6720	0.130
sp ²	2.1370	0.198
<i>Chlorine</i>		
sp ³	2.4140	0.070
<i>Binary Parameter</i>		
H...O	not applic.	6.468

Different types of compounds exhibit different dominant interactions (e.g., hydrogen-bonding dominates the alcohols whereas the dispersion interaction dominates the alkanes). Analysis of individual compound classes is useful to determine the scope and limitations of the model. Mean absolute errors for the individual compound classes are given in Table 6.5.

Table 6.5. Model Accuracy for Different Compound Classes. Results are arranged in order of increasing mean absolute error.

Compound Class	No. Compounds	Boiling Point Range (K)	Mean Abs. Error (K)
nitro	7	484–515	4.6
chloroalkanes	12	285–447	7.6
amines	17	276–560	8.5
OCO-containing	28	304–523	8.7
alcohols	26	351–563	9.0
nitriles	6	354–564	9.4
aldehydes	8	294–376	10.1
ethers	8	248–426	10.8
hydrocarbons	23	231–529	11.1

Several observations can be made on the basis of Table 6.5. The fact that the hydrocarbons, consisting of alkanes, alkenes, and benzenoid hydrocarbons, exhibit a larger error than the polar compounds reflects the dominance of dispersion interactions on these compounds. In fact, correlation of the hydrocarbons' normal boiling points with the surface area alone yields as good or better a fit. A second compound class where significant deviations are observed is the aldehydes. The results indicate that the proposed model form does not predict the charge-transfer complexes and dimers formed by many of these systems. The deviations exhibited by the ether compounds indicate that the model would benefit by differentiating between the sp^3 oxygens in ether and in the hydroxyl groups.

6.5.2. Comparison With Method of Joback

It is well-known that methods based on additive group contributions have been successful in the prediction of pure component properties. Such models assign statistically fitted parameters to different functional groups. The estimation of a particular property is accomplished by summing the contributions for the functional groups present in the molecule. Due to the simplicity of the group contribution approach as compared to the approach presented here, it is desirable to determine whether the proposed model offers improved accuracy. The proposed model in fact shows a reduced mean absolute error as compared to the method of Joback, which has a mean absolute error of $12.1K$ (ca. 3.1%) and a coefficient of determination (r^2) of 0.95. A comparison on the mean absolute errors for the individual compound classes is given in Table 6.6.

Table 6.6. Breakdown of Model Fit and Joback's Predictions into Individual Compound Classes. Arranged in order of increasing mean absolute error.

Compound Class	No. Data	Mean Abs. Error (K) Joback's Method	Mean Abs. Error (K) Proposed Model
nitro	7	39.7	4.6
chloroalkanes	12	11.5	7.6
amines	17	16.8	8.5
OCO-containing	28	5.9	8.7
alcohols	26	12.9	9.0
nitriles	6	10.3	9.4
aldehydes	8	7.0	10.1
ethers	8	10.6	10.8
hydrocarbons	23	10.2	11.1

In most of the categories the proposed model yields better results. A particular limitation of Joback's method that is evident is the unusually large mean absolute error for aromatic nitro compounds (if these compounds are removed the proposed model still has reduced variance). Whereas the model developed in this work was tailored primarily to nitro groups on aromatic substrates, non-aromatic systems are apparently the focus in Joback's model development. The main area where Joback's method is considerably better are for molecules containing carbonyl (sp^2) oxygens. This result is not surprising as the electron density measure is expected to be very limited in treating the underlying interactions.

An important result in the comparison with Joback's method is the reduced number of parameters of the proposed model. For the compounds considered, Joback's method requires 34 parameters (25 were explicitly used in the compounds considered), whereas the model requires only 19 parameters (14 charge scaling factors and 5 linear regression coefficients). Such a reduction in parameters accomplishes one of the major goals of the model building strategy (chapter five); namely, the use of quantum calculations to reduce the total number of adjustable parameters and hence, in principle, reduce the amount of model compound data required to achieve a given statistical significance in the model fit.

The statistical significance of the difference between the mean absolute errors can be established by ANalysis of Variance (ANOVA). Denoting the mean absolute error of the proposed model MAE_{model} and of Joback's method MAE_{Joback} , one-factor ANOVA (Schmidt and Launsby, 1992) is performed to assess whether the two values are different to a statistically significant degree. If MAE_{model} and MAE_{Joback} are statistically discernable, then it can be argued that the proposed model yields a better fit to the data. In the ANOVA formalism two cases are posed: the null hypothesis (H_0) and alternate hypothesis (H_j).

$$H_0 : MAE_{model} = MAE_{Joback}$$

$$H_1 : MAE_{model} \neq MAE_{Joback}$$

Given these null and alternate hypotheses, the steps in ANOVA are presented in Figure 6.5.

Figure 6.5. ANOVA to Determine Significance of Electron Density Parameters.

Step 1: Determine the F-ratio, F_0 , as the ratio of the mean square between, MSB , and the mean square error, MSE . The term MSE is given by

$$MSE = \frac{(n_{model} - 1)S_{model}^2 + (n_{Joback} - 1)S_{Joback}^2}{(n_{model} + n_{Joback} - 2)},$$

where n_{model} and n_{Joback} are the sample sizes (here, $n_{model} = n_{Joback} = 135$) and S_{model}^2 and S_{Joback}^2 are the variances associated with MAE_{model} and MAE_{Joback} . For the case considered,

$$S^2 = \frac{\sum_{m=1}^{n(=135)} [AE_m - MAE]^2}{n - 1},$$

where AE_m is the absolute error of the measurement m . The term MSB is given by

$$MSB = \frac{(n_{model} + n_{Joback})}{4} [MAE_{model} - MAE_{Joback}]^2.$$

In the current example, $S_{model}^2 = 47.3$, $S_{Joback}^2 = 142.1$, $MSE = 94.7$, $MSB = 621.0$, and $F_0 = 6.56$.

Step 2: Obtain the appropriate value from the F-distribution (usually through table lookup) defined by $F_c = F(\beta, 1, n_{model} + n_{Joback} - 2)$, where β is the desired confidence level (e.g., 0.9, 0.975, 0.99, etc.). In the present example, $\beta = 0.975$ is chosen, yielding $F_c \in [5.15, 9.02]$.

Step 3: If $F_0 \leq F_c$, fail to reject H_0 . If $F_0 > F_c$, accept H_1 with a confidence level of $\beta \times 100\%$. In the present example, $6.56 > 5.15$, thus the alternate hypothesis (that the mean absolute errors of the proposed model is less than that for Joback's method) is accepted with a confidence level better than 97.5%.

Having established the statistical significance of the model parameters relative to an independent model with a similar number of parameters, it is next of interest to perform a test-of-fit to determine the robustness of the model parameters to the choice of the dataset.

6.5.3. Test-of-Fit for the Model

In order to perform a test-of-fit analysis, 40 compounds have been removed at random from original 135 to form a training set of 95 compounds (Table 6.7).

Table 6.7. Set of 40 Compounds Removed to Perform Test-of-Fit.

1, 6, 10, 14, 16, 19, 22, 27, 31, 35,
37, 41, 44, 52, 54, 62, 66, 69, 72, 76,
79, 82, 86, 91, 94, 98, 103, 105, 110, 113,
119, 121, 131, 135, 139, 145, 147, 153, 157, 160

Based on the optimized charge scaling factors and regression parameters obtained from the training set, normal boiling points are predicted for the remaining 40 compounds. The graph showing the fit to the training set is given in Figure 6.6.

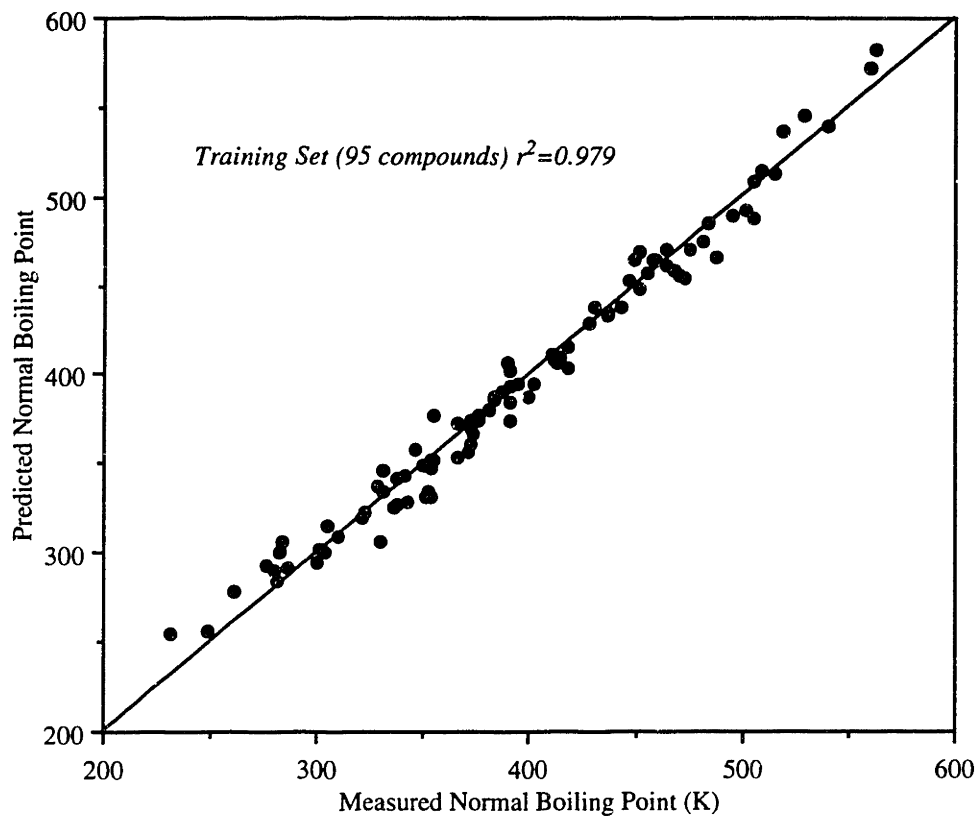


Figure 6.6. Test-of-Fit of the General Model Developed for the Normal Boiling Point (Eq. 21).

The model generated for the training set is next used to predict the normal boiling points for the additional 40 "test-of-fit" compounds excluded from the training set. The resulting fit is shown in Figure 6.7.

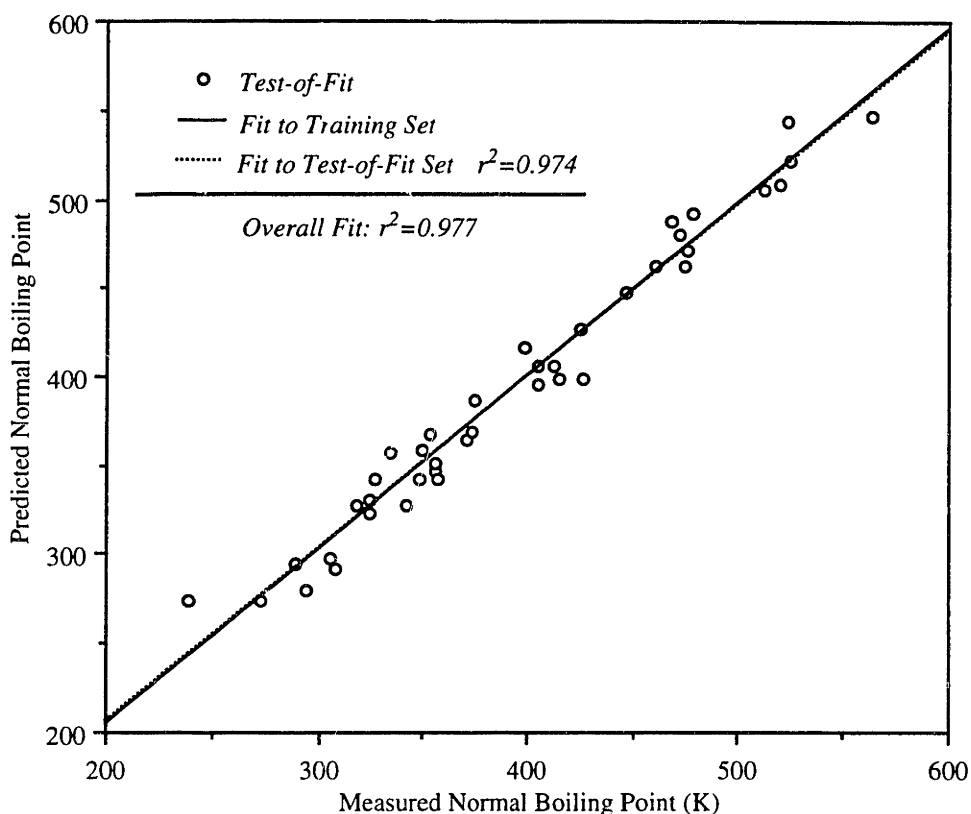


Figure 6.7. Verification of Model Against 40 Compounds Outside the Training Set.

Analysis of the model fit shows that the model created by the training set is essentially indistinguishable from the original model. Figure 6.7 also shows that the fit obtained for the 40 "test-of-fit" compounds is essentially identical to the one generated by the training set. Relevant model statistics are given in Table 6.8.

Table 6.8. Comparison of Model Statistics for the Original Model, the Model Created by the Training Set, and the Fit Obtained for the 40 Test-of-Fit Compounds.

Description of Model	No. Compounds	Mean Absolute Error, K (%)	Coefficient of Determination	Model Parameters				
				dispers.	polar attr.	polar repul.	hyd.-bonding	const.
				$\hat{\alpha}_1$ (*)	$\hat{\alpha}_2$	$\hat{\alpha}_3$	$\hat{\alpha}_4$	$\hat{\epsilon}$
original	135	9.1 (2.4%)	0.977	1.73 (0.04)	-47.2 (.1)	-15.8 (.6)	-31 (2)	101 (6)
training	95	8.3 (2.2%)	0.979	1.76 (0.04)	-47.4 (.1)	-16.6(.6)	-30 (2)	104 (6)
test-of-fit	40	10.6 (2.9%)	0.974	-	-	-	-	-

*Standard Error

6.5.4. Statistical Significance of Electron Density Parameters

While the comparison with Joback's method demonstrates that the model yields comparably good results with a reduced number of parameters, additional tests must be performed to establish the statistical significance of the component pieces of the model. Of particular interest are the electron density parameters. These parameters should be closely scrutinized for several reasons. First, the electron densities are obtained from semi-empirical methods, which can differ considerably from *ab initio* calculations and also may not represent the electron density in the condensed phase. The chlorine and hydrogen atoms, in particular, are subject to considerable deviations (in this regard, it should be noted that empirical modifications can be used to correct the electron density of hydrogen atoms, Grigoras (1990)). Second, there is no unique partitioning of the electron density into partial atomic charges. The Mulliken population analysis used here might, for example, be replaced by alternate methods such as electrostatic potential fitting. Third, the use of partial charges as scaling factors on the hydrogen-bonding terms is suspect since hydrogen-bonding is qualitatively different from the coulombic electrostatic interaction.

In order to evaluate the significance of the electron density parameters, the proposed model form has been compared to the case where all electron density parameters are set to unity. Thus, whereas the model tested has individual contributions of the form

$$(bqp)_{c_1} \cdot (bqp)_{c_2}, \quad (22)$$

it is of interest to compare this form with the reference form created by neglecting the electron density parameters (Eq. 23).

$$(bp)_{c_1} \cdot (bp)_{c_2} \quad (23)$$

The fit obtained using the reference form has a coefficient of determination of 0.972 (as compared to 0.977 for the proposed model) and mean absolute error 10.3K (ca. 2.8%) (as compared to 9.1K for the proposed model). The fit incorporating electron density parameters yields the better estimate of the normal boiling point in 63% of 135 compounds. The coefficients of determination and mean absolute errors, however, are not sufficient to assert the superiority of the model that incorporates electron density parameters for all classes of compounds considered (*e.g.*, an ANOVA analysis of the mean absolute errors for the entire dataset is not decisive). It should be noted, however, that the proposed model form has been consistently better (if only slightly) in each test considered. For example, the test-of-fit analysis yields $r^2 = 0.974$ for the training set as compared to $r^2 = 0.965$ for the 40 additional compounds.

Significant deviations for several classes of compounds tend to reduce the contribution of other compound classes for which the electron density parameters are statistically significant. To illustrate, Table 6.9 breaks down the model fit into individual compound classes.

Table 6.9. Breakdown of Model Fit into Individual Compound Classes. Arranged in order of increasing mean absolute error.

Compound Class	No. Data	Mean Abs. Error (K) Reference Model*	Mean Abs. Error (K) Original Model	No. of Data where Reference is Better
nitro	7	5.6	4.6	2
chloroalkanes	12	8.3	7.6	5
amines	17	12.5	8.5	4
OCO-containing	28	12.7	8.7	7
alcohols	26	8.0	9.0	17
nitriles	6	7.1	9.4	3
aldehydes	8	6.6	10.1	5
ethers	8	10.2	10.8	3
hydrocarbons	23	12.8	11.1	5

*Model in which all partial charges are set to unity.

Furthermore, in the aldehyde and alcohol series, it is observed that fixed parameters rather than variable electron densities tend to yield better predictions. Therefore, vacuum electron density parameters computed by the AM1 method are significantly limited in their ability to represent strong hydrogen-bonding or other interactions involving charge-transfer.

6.6. Transfer of Parameters to Other Properties

Chapter five as well as the work of Grigoras (1990) has demonstrated the transfer of the charge scaling factors from the normal boiling point to the critical temperature (and *visa-versa*) and other pure component properties, such as the critical pressure and volume. It has also been shown that correlation of the critical volume yields transferable atomic radii. Based on a knowledge of any two (or three) of the pure component properties, the remaining properties can be obtained using corresponding state and related models (Reid *et al.*, 1987). The extension to mixture properties, however, requires the explicit development of appropriate models that incorporate the proposed interaction potential in some form.

Ideally, one would like to handle interactions between dissimilar molecules directly (recall that the proposed lattice potential is based on the form of UNIQUAC). Full exploration of such an approach is a major piece of research that will not be undertaken here. Instead, we will test the transferability of the net intermolecular potential energy contributions derived for the pure components.

6.6.1. Octanol-Water Partition Coefficients of Nitroaromatics

The first property to be investigated is the octanol-water partition coefficient, defined as the equilibrium constant of partitioning of a solute between an octanol phase and a water phase at 25°C (Eq. 24).

$$P_{ow}(m) = \frac{C^o(m)}{C^w(m)}, \quad (24)$$

The octanol-water partition coefficient is widely used as an indicator in environmental and toxicity studies. In this section the intermolecular potential descriptors are utilized to predict the partition coefficients for a class of nitroaromatics.

A relationship between the partition coefficient and the intermolecular potential follows readily from the linear free-energy treatment. Denoting octanol by "o" and water by "w", the partition coefficient at constant pressure and temperature is logarithmically proportional to the molar Gibbs free energy difference between the two phases, as shown in Eq. 25.

$$RT \log P_{ow}(m) = a \cdot \Delta_{ow}G(m), \quad (25)$$

where

$$\Delta_{ow}G(m) \equiv G_o(m) - G_w(m) \quad (26)$$

and a is a constant of proportionality. Following the standard formalism of the linear free energy treatment, Eq. 25 is augmented by a reference system to yield Eq. 27.

$$RT \log \frac{P_{ow}(m)}{P_{ow}^r(m)} = a \cdot (\Delta_{ow}G(m) - \Delta_{ow}G^r(m)), \quad (27)$$

where P_{ow}^r and $\Delta_{ow}G^r(m)$ describe the reference system. Using the definition of the Gibbs free energy allows a separation of Eq. 27 into the molar enthalpic and entropic components (Eq. 28).

$$RT \log \frac{P_{ow}(m)}{P_{ow}^r(m)} = a \cdot [\delta\Delta_{ow}H(m) - T\delta\Delta_{ow}S(m)], \quad (28)$$

where

$$\delta\Delta_{ow}H(m) \equiv \Delta_{ow}H(m) - \Delta_{ow}H^r(m), \quad (29)$$

and

$$\delta\Delta_{ow} S(m) \equiv \Delta_{ow} S(m) - \Delta_{ow} S(m). \quad (30)$$

For a class of compounds of similar size, intermolecular potential, and flexibility, the combinatorial entropy difference (Eq. 30) between each compound and the reference compound (also from the same class) might reasonably be assumed constant or, at most, proportional to Eq. 29:

$$T \delta\Delta_{ow} S(m) \propto \delta\Delta_{ow} H(m). \quad (31)$$

Under this assumption, Eq. 28 may be simplified to the form given in Eq. 32.

$$RT \log \frac{P_{ow}(m)}{P_{ow}(m)} = b \cdot \delta\Delta_{ow} H(m) + c, \quad (32)$$

Utilizing the definition of the enthalpy ($\Delta H = \Delta E - P\Delta V$) with further assumption that the volume change between the reference molecule and the molecule of interest is negligible, *i.e.*,

$$\delta\Delta_{ow} V(m) \text{ ca. constant}, \quad (33)$$

it follows that Eq. 32 reduces to Eq. 34.

$$RT \log \frac{P_{ow}(m)}{P_{ow}(m)} = b \cdot \delta\Delta_{ow} E(m) + c, \quad (34)$$

where b and c are constants. As in the case of the boiling point, canceling the kinetic and intramolecular contributions to the internal energy yields an approximate linear relationship between the property of interest (in this case, the partition coefficient) and the intermolecular potential. As a first approximation, one can attempt to account for the difference between the two solvent environments by fitting linear weighting coefficients to the pure-component intermolecular potential contributions of the solute, *i.e.*,

$$\log P_{ow}(m) = \alpha_1 f_1(m) + \alpha_2 f_2(m) + \alpha_3 f_3(m) + \log P_{ow}, \quad (35)$$

where $\log P_{ow}$ is essentially a constant associated with the reference compound. The hydrogen-bonding contribution is not included in Eq. 35 since the compounds under consideration do not have a special hydrogen-bonding contribution.

In order to test Eq. 35, a set of eleven nitroaromatics are considered. Computational results yield a coefficient of determination *ca.* 0.97. The model parameters obtained are given in Table 6.10.

Table 6.10. Model Parameters and Statistics Obtained for the Prediction of the Octanol-Water Partition Coefficient.

Mean Absolute Error (K)	Coefficient of Determination	Model Parameters			
		dispers.	polar attr.	polar repul.	const.
		$\hat{\alpha}_1(^*)$	$\hat{\alpha}_2$	$\hat{\alpha}_3$	$\log P_{ow}$
0.24 (12.8%)	0.97	0.028 (0.003)	-0.15 (0.1)	0.40 (0.03)	-7 (1)

*Standard Error.

One of the original reasons for studying nitroaromatics has been to determine whether polar descriptors can be used to predict the partition coefficient, or whether more general treatment of hydrogen-bonding is required (Murray *et al.*, 1993). Judging from Table 6.10, the most significant parameters are the dispersion and polar repulsive interactions. Murray *et al.*, (1993) obtained a similar effect using parameters derived from the electrostatic potential and attributed it to the positive potentials associated with the nitro groups. The results given in Table 6.10 are generally in accord with those of Murray *et al.* (1993) and demonstrate that polar parameters are capable of yielding reasonable approximations for this class of compounds (although the result probably does not hold for a dataset containing different classes of molecules). The specific predictions are given in Table 6.11.

Table 6.11. Report of Experimentally Determined and Predicted Octanol-Water Partition Coefficients for 11 Compounds Related to Nitroaromatics. Measured coefficients from Murray *et al.* (1993).

Solute	Compound Databank Entry	Partition Coefficient		Interaction Potential Descriptors		
		$\log P_{ow}^{exp}$	$\log P_{ow}^{pred}$	dispers.	polar attr.	polar repul.
benzene	1	2.13	2.43	159.8	-3.51	11.04
toluene	7	2.73	2.80	183.7	-2.89	10.55
nitrobenzene	2	1.85	1.95	192.8	-3.64	7.49
2-nitrotoluene	60	2.30	2.46	210.0	-3.21	7.73
3-nitrotoluene	61	2.45	2.49	217.5	-3.16	7.30
4-nitrotoluene	62	2.42	2.55	216.9	-3.20	7.47
1,3-dinitrobenzene	171	1.49	1.55	226.4	-3.09	4.35
2,4-dinitrotoluene	172	1.98	2.17	243.8	-2.97	4.73
2,6-dinitrotoluene	173	2.02	2.29	238.4	-3.19	5.31
1,3,5-trinitrobenzene	174	1.18	1.37	258.0	-2.03	2.10
2,4,6-trinitrotoluene	175	1.86	2.04	270.6	-2.39	2.75

The quality of the predictions compare reasonably well with those obtained using electrostatic potential-derived descriptors (Murray *et al.*, 1993). The proposed model does

not do as a good job on the acetonitrile-water partition coefficient (coefficient of determination *ca.* 0.91, nor does it reproduce the partition coefficient for a data set containing a wide variety of different classes of compounds (such a dataset is given in Brinck *et al.*, 1993). The limited capability to handle a wide range of different compounds can be tied in part to the entropy and volume assumptions given above (Eqs. 31 and 33). A more general limitation is that the specific interactions between the solvents and the solute are not being computed.

6.6.2. Linear Solvation-Energy Relationship for the Menshutkin Reaction

In addition to the treatment of solute effects, there are a number of important properties which involve the treatment of solvent effects. The second property that is investigated is a linear solvation-energy relationship, which can be used to predict the effects of different solvents on reaction rates. Consider a bimolecular reaction of the form given in Eq. 36, which assumes a single activated complex.



The choice of solvent can increase or decrease the reaction rate by increasing or decreasing the free energy of solvation of the transition state with respect to the reactant molecules. Such an effect is amenable to transition state theory. Transition state theory allows that the ratio of rates between a solvent and a reference solvent is given by Eq. 37.

$$\log \frac{k}{k'} = -\frac{\Delta G_{solv}}{RT}, \quad (37)$$

where ΔG_{solv} is the difference in the net free energy of solvation given by Eq. 38.

$$\Delta G_{solv} = \Delta G_{A,solv} + \Delta G_{B,solv} - \Delta G_{AB^{\ddagger},solv} \quad (38)$$

Equation 38 shows that a key challenge in predicting the solvent effect on the relative reaction rate is to model the thermodynamic interactions between the solvent and the transition state and between the solvent and the reactants. Current approaches to linear solvation-energy relationships are limited on several grounds. Correlation of solvachromatic parameters, for example, requires direct measurements of the parameters on the solvents of interest (Reichardt, 1990), whereas in principle it would be desirable to look at novel solvents. Thus, in order to propose a new model it is necessary to perform a large number of direct measurements on the solvents of interest. Due to the wide variety of solvent effects and the different models employed, availability of a set of transferable parameters for the underlying atomic contributions is highly desirable. At the same time, accounting for the structural dependency of the electron distribution is essential if we are to

look at reaction mechanisms involving considerable charge separation in the reacting substrate.

The set of reductions allowing simplification of Eqs. 37 and 38 to an approximate linear relationship with the intermolecular potential is identical to those used to derive Eq. 34. Therefore, an equation similar to Eq. 35 is proposed for the linear solvation energy relationship, namely Eq. 39.

$$\log k(m) = \alpha_1 f_1(m) + \alpha_2 f_2(m) + \alpha_3 f_3(m) + \log k \quad (39)$$

Equation 44 has been tested for the prediction of solvent effects in a Menshutkin reaction (Figure 6.8), which has been used to test numerous linear-solvation energy relationships (Reichardt, 1990).

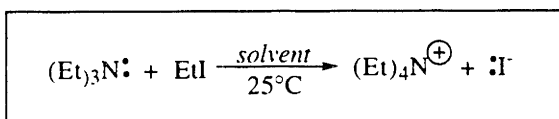


Figure 6.8. Depiction of the Menshutkin Reaction Used to Test the Proposed Linear Solvation-Energy Relationship.

A set of twelve compounds has been selected that includes nitriles, chlorinated hydrocarbons, nitro compounds, and aldehydes. The charge scaling factors are transferred from the normal boiling point as before, but for a special subset of the compounds in the databank (described below). The initial fit to the dispersion, polar attractive, and polar repulsive contributions yields a coefficient of determination *ca.* 0.936 (Table 6.12).

Table 6.12. Model Parameters and Statistics Obtained for the Linear Solvation-Energy relationship.

Description of Model	Mean Absolute Error (%)	Coefficient of Determination	Model Parameters			
			dispers.	polar attr	polar repul.	const.
			$\hat{\alpha}_1$ (*)	$\hat{\alpha}_2$	$\hat{\alpha}_3$	$\log k$
includes disp.	0.53(16%)	0.936	-0.001 (0.01)	-2.5 (0.3)	-1.0 (0.1)	4.6 (1.9)
no disp.	same	0.936	-	-2.6 (0.2)	-0.96 (0.09)	4.4 (0.4)

*Standard Error

Inspection of the standard errors of the coefficients reveals that the dispersion contribution (*i.e.*, the molecular surface area) is statistically insignificant. The outcome is not surprising and is actually desirable since the reaction mechanism involves an ionic complex, wherein the most important contribution to solvation should arise from ion-

solvent dipole interactions. A list of compounds and the computational predictions are given in Table 6.13.

Table 6.13. Report of Experimentally Determined and Predicted Reaction Rates for the Menschutkin Reaction in 12 Selected Solvents. Original data from Abraham et al., (1976). Compounds related to nitroaromatics. Measured coefficients from Murray et al. (1993).

Solute	Compound Databank Entry	Reaction Rate*		Interaction Potential Descriptors		
		$\ln 10^{\delta} k^{exp}$	$\ln 10^{\delta} k^{pred}$	dispers.	polar attr.	polar repul.
benzene	1	1.68	2.46	159.8	-3.59	11.78
benzotrile	5	5.02	4.74	194.9	-3.67	9.65
chlorobenzene	6	2.97	3.26	181.7	-2.31	7.47
toluene	7	1.21	1.32	183.7	-2.94	11.22
hexane	27	-4.31	-4.09	211.0	0.00	8.86
chloroform	31	3.40	4.48	150.8	-0.09	0.18
1,1-dichloroethane	36	3.17	3.64	152.2	-0.37	1.81
1,2-dichloroethane	37	4.54	3.94	154.8	-0.50	1.87
nitromethane	57	5.81	6.11	120.3	-3.57	7.94
acetone	133	4.18	3.21	138.3	-1.39	5.04
butan-2-one	134	3.68	2.47	159.5	-1.40	5.84
nitrobenzene	2	5.21	5.16	192.8	-6.68	17.36

*units for rate constant: $1 \text{ mol}^{-1} \text{ sec}^{-1}$.

A plot of the predicted versus the measured rate constants is revealing. While the overall fit is reasonable (Figure 6.9), there are systematic errors in the predicted rate constants that correspond roughly to the different classes of compounds in the dataset. Three distinct sub-classes of compounds seem to exhibit systematic deviations: (i) compounds containing an sp^2 -oxygen such as nitroaromatics and aldehydes; (ii) chlorinated aromatics; and (iii) aromatic hydrocarbons. The presence of such systematic errors is believed to result from the fact that the interaction parameters deal only with the solvent and as a result the special interactions associated with each compound class are averaged out in the correlation, resulting in systematic deviations.

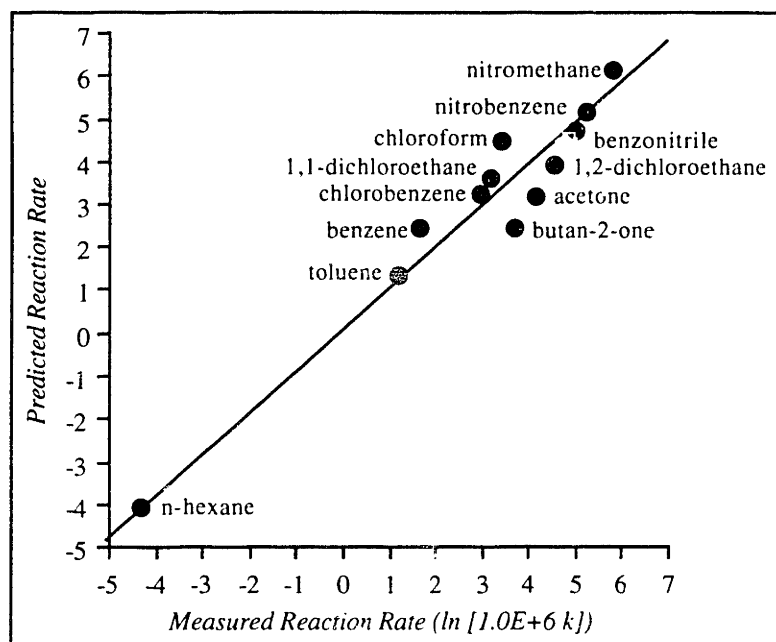


Figure 6.9. Correlation of the Reaction Rate and a Function of the Choice of Solvent for the Menshutkin Reaction. Units for the rate constant are in $l\ mol^{-1}\ sec^{-1}$.

The charge scaling factors used to compute the polar interaction parameters have been obtained from a subset of the compounds used in computing the original model for the normal boiling point. The compound classes included are the hydrocarbons (with the exception of alkene and allenes), chlorinated alkanes, nitriles, nitro compounds (including nitromethane), aldehydes, and amines. A fit of the normal boiling point to the interaction potential contributions yielded a considerably better fit than for the compound class as a whole, resulting in a coefficient of determination of 0.988 (ca. 2.0%). Table 6.14 summarizes the relevant model statistics.

Table 6.14. Model Parameters and Statistics for the Dataset used to Obtain Parameters for the Linear Solvation-Energy Relationship.

Mean Absolute Error (K)	Coefficient of Determination	Model Parameters				
		dispers.	polar attr.	polar repul.	hyd.-bonding	const.
		$\hat{\alpha}_1$ (*)	$\hat{\alpha}_2$	$\hat{\alpha}_3$	$\hat{\alpha}_4$	$\hat{\epsilon}$
7.4 (2.0%)	0.988	1.91 (0.05)	-46.7 (1.0)	-13.9 (0.6)	-2.79 (2.32)	55 (8)

*Standard Error.

The set of optimized charge scaling factors is given in Table 6.15.

Table 6.15. List of Input Model Parameters Used in Linear Solvation-Energy Relationship.

Component	Optimized Radius (Å)	Charge Scaling Factor (Å)
<i>Hydrogen</i>		
aliphatic	1.6978	1.683
sp ²	1.8032	1.796
hydrogen-bonded	1.3961	-1.103
<i>Carbon</i>		
aliphatic	1.6930	1.000
aromatic	1.9411	0.987
sp	2.4980	2.860
<i>Nitrogen</i>		
sp ³	1.9767	-0.801
sp ²	2.3360	2.841
sp	2.4120	-1.613
<i>Oxygen</i>		
sp ²	2.1370	0.335
<i>Chlorine</i>		
sp ³	2.4140	0.199
<i>Binary Parameter</i>		
H••N	not applic.	1.150

Notice that the weak hydrogen-bonding due to amines is accounted for in the model. This binary parameter is included to test the earlier assertion that the resulting interaction potential contribution does not improve the fit. Table 6.14 clearly shows that the corresponding parameter, $\hat{\alpha}_4$, is statistically insignificant.

A basic conclusion drawn from the development of the linear solvation-energy model is that improved transfer of parameters can be attained by correlating the normal boiling point (or whichever underlying property is being used) with the specific compound classes of interest for the final property. Hence, one way to refine the model is to refine the charge scaling factors in the first place. Such a capability seems essential if the full potential of developing transferable model parameters is to be achieved.

6.7. Discussion

This chapter has demonstrated the potential for an integrated chemical modeling strategy in which (i) intermolecular potential-related parameters are derived from readily available properties such as the normal boiling point; and (ii) approximate theories are used to gain insight into additional properties to which the interaction parameters can be transferred. A limiting feature of the initial treatment has been that the intermolecular potential contributions are computed only for pure component systems. While being directly applicable in estimating the lattice potential of a system consisting of a single compound, extension to mixture properties requires a more empirical treatment of the lattice potential.

As long as the mixture properties are for a narrowly defined class of compounds (*e.g.*, the partition coefficients of the nitroaromatics), the use of pure component descriptors can yield reasonable results because the special types of interactions between the solute (or solvent) and the other mixture components are relatively uniform and can be incorporated into the regression parameters. When a broader set containing several classes of compounds are considered, however, the approach breaks down. The failure occurs because the regression parameters effectively average-out important intermolecular potential contributions that distinguish between the individual classes.

The failure to account for diverse datasets is one reason motivating the explicit treatment of dissimilar molecules directly in the intermolecular potential. A second reason is that the linear property-energy formalism does not work well for many properties, due to failures in the constant entropy assumption. Thus, while it may be useful to obtain the charge scaling factors (or whatever intrinsic parameters are being derived), ultimately these parameters must be incorporated *via* the lattice representation of the intermolecular potential into more general statistical thermodynamic models such as UNIFAC.

A basic advantage of the proposed approach for use in UNIFAC or other thermodynamic models, assuming the parameters can be generalized to better represent hydrogen-bonding and charge-transfer interactions, is that it can handle molecules and ions with considerable charge separation *via* the electron density, isodesmeric reaction sequences, or other molecular parameters. Thus, the approach may provide a basis for extending general purpose statistical thermodynamic models to larger class of applications, including linear solvation-energy relationships. As a first step towards demonstrating the potential advantages of models that incorporate the electron density, one might consider developing prototype pure component thermophysical property packages such as those used in process simulation tools. Demonstration that such models can predict separation factors between two molecules (*e.g.*, polysubstituted aromatic isomers) where traditional methods would predict equal vapor pressures should provide strong evidence for pursuing the approach further. This demonstration is accomplished in chapter nine.

6.8. Conclusions

Computational experience with the chemical modeling strategy proposed in chapter five has been reported. A model form has been proposed and results for correlation of the normal boiling point reported. Comparisons with Joback's method show that the proposed method is generally more accurate and involves approximately 25% fewer adjustable parameters. Thus, the generalization of the geometric representation accomplished by (i) using molecular orbital geometry optimizations to obtain a reference geometry; (ii) correlation of the atomic radii with the critical volume; and (iii) numerical integration to determine the molecular surface area of individual atom-type contributions is an effective strategy.

Initial progress has been reported in transferring the charge scaling factors computed for pure component properties to additional properties. Transfer to critical properties has been reported in chapter five. Additionally, transfer of parameters to two mixture properties has been accomplished in this chapter, one in which the pure component contributions represented the solute (*i.e.*, the partition coefficient) and one in which the contributions represent the solvent (*i.e.*, the linear solvation-energy relationship). Two key requirements for an effective strategy have been demonstrated. First, it has been shown that model refinement in a final desired property can be accomplished through refinement in the dataset used to obtain the charge scaling factors. Second, the transferability to multiple properties of interest has been demonstrated.

The scope and limits of this approach within a linear free-energy modeling framework have been discussed. Evidence has been presented that extension to direct incorporation of intermolecular interactions between dissimilar molecules should considerably generalize the method. This extension and the improvement in the intermolecular potential model to better represent hydrogen-bonding and charge-transfer complexes (most likely through isodesmic reaction sequences) are important next steps.

Chapter 7

Integration of Reaction Path Selection with Process Design Decisions. A Total Plant Synthesis Approach.

7.1. Introduction

Three key levels in process design are

- (i) selection of chemical reaction paths;
- (ii) development of appropriate unit operations; and
- (iii) design of the process flowsheet.

Most process synthesis tools address the design of process flowsheets or related subproblems. The design space is generally represented by a map of possible configurations of unit operations, each of which is described by a set of equations or by a simulation module. A general introduction to systems engineering techniques is given in appendix three. Much less work has been done to address tasks (i) or (ii). These tasks involve physical processes that are difficult to model, require continual introduction of new experimental information, and involve complex organizational issues. As a result, the general process synthesis problem (i)—(iii) is regarded as both difficult to characterize and computationally overwhelming (Fornari, 1989).

There remains, however, a considerable need to systematically address specific aspects of tasks (i) and (ii). In particular, while current approaches to development of process chemistry are capable of mapping out promising reaction paths satisfying targets of a general nature (*e.g.*, minimum acceptable yield and selectivity of a reaction), there is currently no systematic way to account for many relevant process objectives and production constraints that should favor a particular reaction path(s) configuration over others. Appropriate process synthesis tools could potentially help to

- identify economic-environmental, environmental-safety, and other types of tradeoffs present in the choice of process chemistry;
- develop a more quantitative basis for interaction with regulatory bodies;
- utilize intermediate data and results in experimental design; and
- incorporate probability distribution information in decision-making.

One angle of attack to obtain a better characterization of these issues is the systematic cross-comparison of design alternatives in tasks (i) and (ii) with higher level objectives in task (iii). An improved ability to compare alternate chemical reaction paths developed in the laboratory would be particularly useful. Distinct tradeoffs are already implicit in lab-scale chemistry. For example, two significant tradeoffs are

- reduced waste by separation and recovery of by-products versus increased separations costs; and
- improved economy-of-scale through the development of strategic raw materials and intermediates versus reduced process flexibility to respond to changing raw material and product prices.

A major challenge in addressing such tradeoffs is to represent the relevant attributes of the process chemistry without becoming bogged down by information limitations associated with the choice of unit operations in the flowsheet. To frame the problem, it is useful to consider aggregate representations of a chemical process. In chapter two, a chemical conversion-based representation has been developed in the context of reaction path synthesis. An extension of the chemical conversion concept is the unit process concept. In a unit process, the individual unit operations associated with a chemical conversion are aggregated into net reactions, stabilization requirements, separations requirements, and so on.

This chapter proposes a combinatorial design strategy that exploits mixed-integer linear programming for the preliminary screening of unit process-based networks. The goal of the mathematical programming formulation is to systematically treat synergisms and tradeoffs such as those described above in the selection of reaction paths. As mentioned previously, the set of reaction paths is captured through a reaction path network representation—*i.e.*, a network of interconnected reaction path alternatives for a series of related products. Reaction path networks can be present within the boundaries of a single chemical process (*e.g.*, a multipurpose plant manufacturing series of related products) or may be characteristic of an industry (*e.g.*, a set of petroleum derived chemicals) depending on the relevant economy of scale. Often the tradeoffs of interest are apparent in the topology of the network. For example, the simple reaction path network given in Figure 7.1 illustrates several possible configurations that can be generated through the choice of different raw materials, products, and separations.

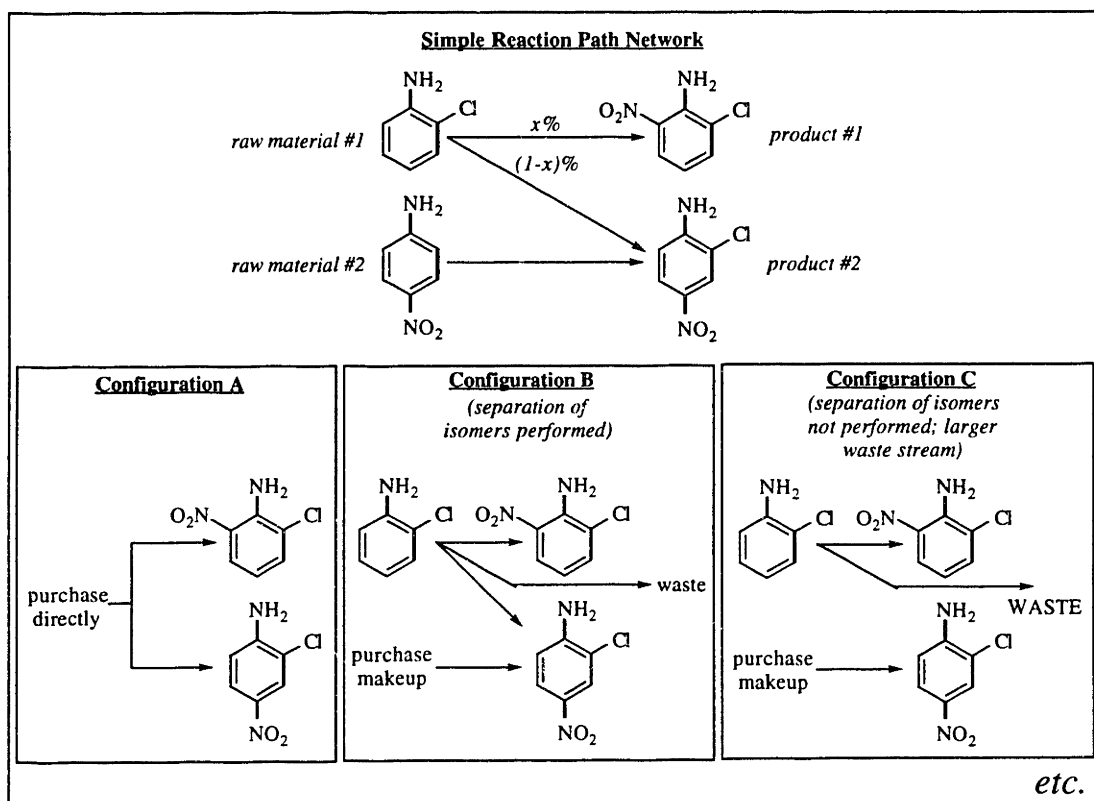


Figure 7.1. Depiction of a Reaction Path Network and Several Configurations Which Indicate Possible Tradeoffs in the Choice of System Chemistry.

In particular, configurations B and C illustrate a possible tradeoff between the separations requirements for recovery of the isomer and the size of the waste stream. From the surprisingly large number of possible configurations that can be generated by even a simple network such as Figure 7.1, it is clear that design tradeoffs present at specific locations in the network act as large multipliers on the total number of candidate processes. Thus, the reaction path network provides a good test of the performance of the combinatorial design strategy.

The chapter begins with a conceptual formulation of the design problem. Important in the formulation is the method of production (*e.g.*, continuous or batch), storage assumptions, and the relationship between equipment sizes and time-averaged production rates. Next, a mixed-integer linear optimization problem is derived optimizing the net present value of the unit process network taking into account variable topology within unit processes. Finally, a chance-constrained mathematical programming formulation is derived for treatment of probabilistic uncertainty in production targets.

7.2. General Strategy

The general strategy pursued in this work is to recast a reaction path network as a flowsheet synthesis problem in which unit operations are aggregated into a set of unit processes, each corresponding to an individual chemical conversion. In particular, a partitioning of reaction paths into unit process models and storage element models is sought (Figure 7.2).

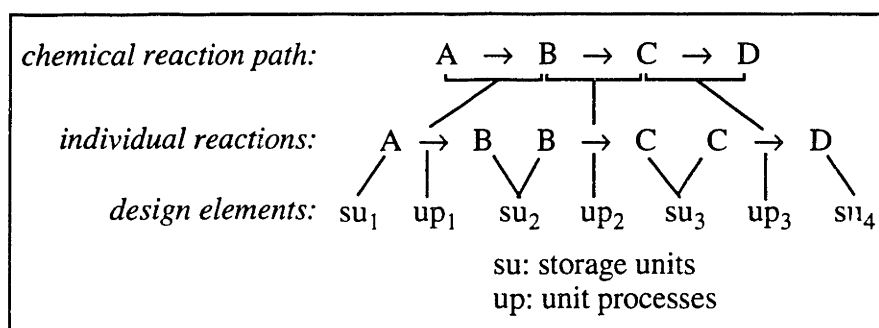


Figure 7.2. Partitioning of a Three Step Reaction Path Into Storage Units and Unit Processes.

In developing the mathematical representation, it is important to appreciate that information sources available for screening a chemical reaction path generally fall short of a complete specification of process elements and production targets. For example, available data on reaction rates and thermodynamic properties required to design equipment may cover only a few specific operating conditions. Furthermore, a proposed process configuration is likely to be subject to considerable modification over time. In order to bridge the gap, the unit process representation almost invariably requires significant assumptions about the choice of underlying unit operations. The approach taken in this work is to introduce explicit study-level design estimates consistent with the available level of information (Figure 7.3).

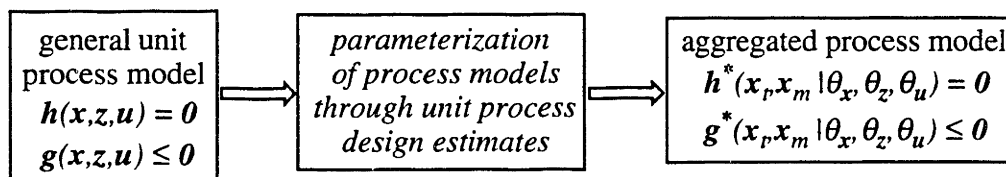


Figure 7.3. Introduction of Design Estimates to Simplify a General Process Model.

Resulting unit process models, $(h^* = 0, g^* \leq 0)$ are essentially parameterizations of the design estimates, consisting of a set of topological alternatives, x_p , with one or more material, energy, and equipment factors, x_m . These factors are responsible for scaling the

overall design to meet required production rates. Remaining design (x), control (z), and state (u) variables are fixed through the design estimates as discrete parameters ($\theta_x, \theta_z, \theta_u$) in the model. In short, the unit process representations are scalable to different feed sizes and can have variable topology, but have static material conversions factors.

There are two main motivations for the proposed approach. First, process design and simulation tools can be exploited to provide base case design estimates for use in the specification of unit processes. Second, design attributes related to the mode of production (*e.g.*, batch or continuous unit processes) can be considered. From the individual base case design of unit processes, material balances associated with a reaction path can be written at two levels: within individual unit processes (*e.g.*, reagent flows, internal splits); and externally around one or more unit processes or storage elements. In this work, the mathematical formulation of a unit process is for batch processing with continuously sized units whose size factors scale directly with the feed size. As shown below, these assumptions lead to unit processes being scaled using the feed batch size to a unit process, ultimately allowing capital and operating costs to be estimated using piece-wise linear functions.

The proposed unit process model form is termed an *unit process superstructure*. The term "superstructure" is invoked due to the fact that several configurations of unit operations can be incorporated, such as allowing the option of performing the separations required to recover a usable reaction by-product. A graphical depiction of a unit process superstructure is given in Figure 7.4.

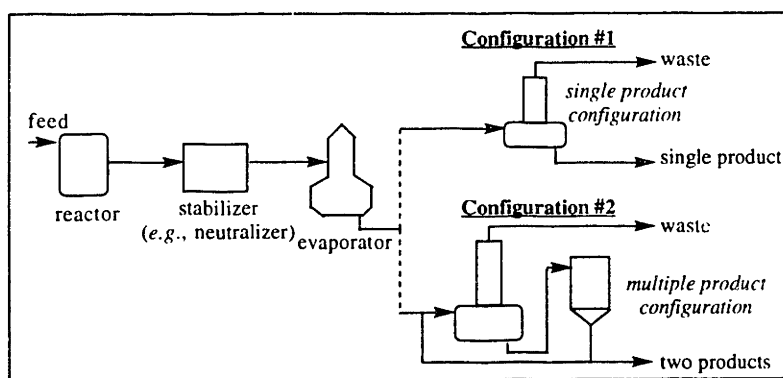


Figure 7.4. Example Unit Process With a Variable Topology Containing Two Separation Schemes.

The first configuration depicts recovery of a single component using batch distillation with disposal of a by-product as waste. The second configuration involves recovery of the by-product. The latter configuration involves both a distillation and a crystallization and requires larger internal process volumes. As indicated above, the internal volumes, heat exchanger surfaces areas, *etc.* can be approximately scaled to different feed sizes. The

material conversion factors for each realization of the unit process are static, however, since they are fixed by base case design estimates.

Through parameterization of unit process models with design estimates, it is computationally tractable to consider integration of chemical reaction paths on several fronts, including (i) process subsystems such as solvent recovery; (ii) by-product and feed stream-types; and (iii) reaction and separation tasks. Such integration problems can be defined by a three step procedure:

- Step 1:* Expand the network of chemical reaction paths into independent chemical reaction paths.
- Step 2:* Identify common subpaths or other relevant features among chemical reaction paths.
- Step 3:* Develop a mathematical formulation for integrating unit processes and/or stream-types.

The integration problems treated in this chapter consist of merging stream-types which have the same chemical makeup and merging unit processes which have the same set of unit operations. For each unit process it is assumed that only one stream-type is suitable as a feed. As shown in Figure 7.5, the integration problem is formulated by replacing each chemical reaction path with a directed graph composed of (i) a set of nodes representing intermediate storage units; (ii) a set of nodes representing unit processes; and (iii) a network of arcs representing mass flows.

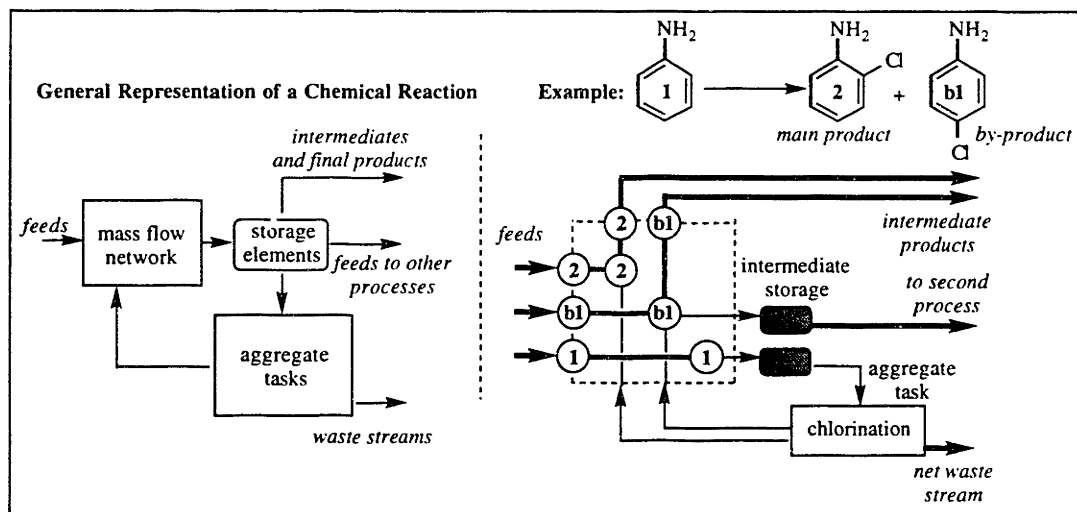


Figure 7.5. Representation of the Mass Flows and Unit Processes for a Single Chemical Reaction in a Chemical Reaction Path.

Several discrete design alternatives are present in Figure 7.5. For example, the recovery of by-product (b1) to intermediate storage represents one design alternative which should lead

to relatively low levels of waste stream generation. Alternately, a second design is to avoid the reaction altogether, instead purchasing the product as a raw material. Whereas the case treated in Figure 7.5 is a simple one, the graph representation in general can include multiple reactions and pathways, giving rise to an intricate set of design alternatives.

The basic task of this chapter is to codify design alternatives present in the graph representation of the process flowsheet. Since the problem considered below allows for the utilization of batch operations, scheduling-related assumptions must play a significant role in the formulation. Several assumptions are present in the mathematical formulation that effectively eliminate the scheduling of equipment and mass flows from the formulation. First, unique unit-to-task assignments are assumed—*i.e.*, there are no equipment limitations. Second, unlimited intermediate storage units are available, avoiding the need to schedule mass flows in a network of chemical reaction paths. As a result of these assumptions, mass balances between unit processes can be defined as averages of the discrete mass flows.

Based on the definition of unit processes and the assumptions related to scheduling, a configuration of chemical reaction paths can be represented as a directed graph containing two types of nodes: (i) nodes representing unit processes which are specific to each chemical reaction; and (ii) nodes representing intermediate storage units which are specific to each stream-type. Each unit process is thus categorized by a pair of integers, $(i, j) \in I \times J$, where I is the set of chemical reaction paths and J is the set which defines the numerical order of steps in a chemical reaction path. As shown in Figure 7.6, when there are no cycles in the chemical reaction paths the directed graph can be partitioned into stages which consist of a set of unit processes connected to a set of intermediate products. In this case, the set J coincides with the stages of unit processes and intermediate storage units. It should be noted that availability of parallel equipment can be incorporated through the replication of unit processes in each stage. A convenient assumption that can be applied consistently is that each distinct chemical reaction path creates a unique sequence of unit processes.

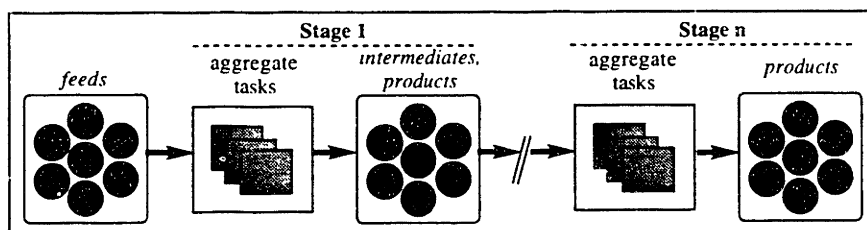


Figure 7.6. Arrangement of the Stream-types and Unit processes Into Stages.

The integration problem defined by this graph can be formulated as a mixed-integer nonlinear programming problem. Continuous variables present in the programming

problem include mass flows and equipment sizes. Discrete variables include choices among raw materials and intermediates, as well as configurations of unit operations. Nonlinear equations present in the programming problem include short-cut equipment cost models, which are often nonconvex. Due to the fact that there are usually no cross products between continuous variables in short-cut models, however, they are usually separable. As described in the next section, separability allows the mixed-integer nonlinear programming problem to be reformulated as a mixed-integer linear programming problem.

7.3. Mathematical Programming Formulation

The design problem posed in the last section can be stated formally as follows: given a set of unit process models $(h_{ij}^* = 0, g_{ij}^* \leq 0, (i, j) \in I \times J)$, a set of storage elements $(\{k\} \equiv K)$, production targets, and any additional constraints, synthesize the optimum process flowsheet for an algebraic design objective, Z . The presence of multiple reaction paths and raw material sources endows the problem with many choices of separation trains, by-products, and waste streams that might satisfy the production targets.

One of the most important set of design tradeoffs to consider are the separations alternatives. The mathematical formulation of these discrete alternatives can be achieved through the introduction of discrete (binary) variables representing the individual choices. Constraints containing both binary and continuous variables can then be derived to express the logical relationships among the design decisions. The resulting total plant synthesis problem then takes the form of a mixed-integer nonlinear program. The basic components of the programming problem are:

- material balances on the individual unit processes;
- material balances on the entire flowsheet;
- equipment design equations;
- production targets;
- constraints on flowsheet structure; and
- an objective function;

Standard assumptions are introduced to simplify the mathematical formulation. First, binary design variables are assumed separable from the continuous variables. Most mathematical forms that involve products of binary and continuous variables can be reformulated using exact linearization techniques. Second, the objective function and constraints are assumed to take a mathematical form that can be linearized.

7.3.1. Flowsheet Material Balances

By appeal to the assumption of unlimited intermediate storage, the time-averaged mass flows can be computed by averaging the discrete mass flows entering and leaving intermediate storage units. A single intermediate storage unit is assigned to each stream-type, $k \in K$, where K is a set of integer indices spanning all stream-types. This assignment implies that each unit process draws and empties into the storage units defined for the appropriate stream-types. Consequently, mass balances (Eq. 1) are written around intermediate storage units for each stream-type in the network.

$$\bar{P}_k - \bar{F}_k = \sum_{(i,j) \in PS(k)} \bar{p}_{ijk} - \sum_{(i,j) \in FS(k)} \bar{f}_{ijk}, \quad k \in K \quad (1)$$

\bar{P}_k = time-averaged net product from process.

\bar{F}_k = time-averaged net feed to process.

$PS(k)$ = set of unit processes producing stream-type k .

$FS(k)$ = set of unit processes using stream-type k as feed.

\bar{p}_{ijk} = time-averaged rate of product k from a unit process (i, j) .

\bar{f}_{ijk} = time-averaged rate of feed flow k to a unit process (i, j) .

7.3.2. Feed Flow Design Equations

Feed flow design equations account for the variation in cycle time as a function of the unit operation configuration. An increase in the cycle time requires that larger equipment volumes be used to process a required amount of material in a given time horizon. Therefore, increased separations to recover useful by-products should be compared against possibly higher equipment and utilities costs. Due to the presence of discrete variables characterizing unit operation configurations, the formulation of these equations should be targeted to improve the computational performance of the algorithm. The required analysis consists of deriving constraints that yield tight linear programming relaxations while keeping the problem size under control. To demonstrate several relevant techniques, the set of net material flows treated by the unit process models depicted in Figure 7.7 is utilized.

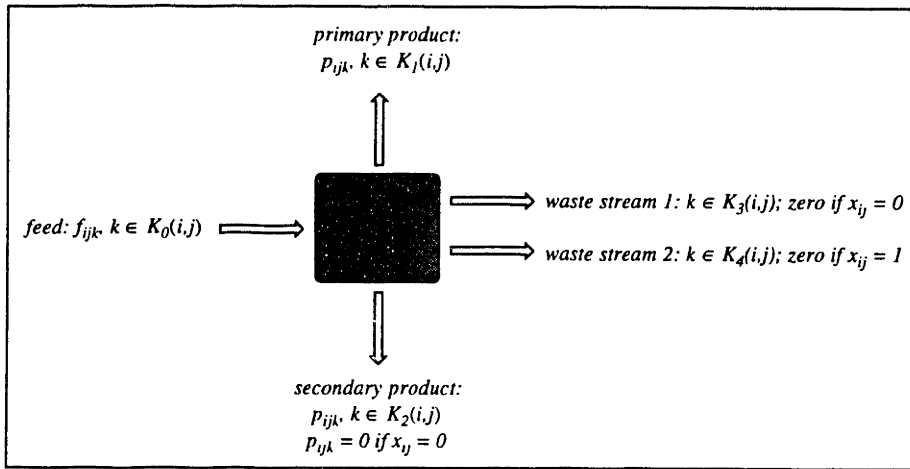


Figure 7.7. Superstructure of Net Material Flows Treated By the Unit Process Models.

As shown in Figure 7.7, each unit process is characterized by a true-false design variable, $x_{ij} \in \{0, 1\}$ where $(i, j) \in I \times J$. If $x_{ij} = 1$, the unit process is modified to recover a second product. In addition to altering the flowsheet, the recovery of a secondary product also increases the nominal cycle time, t_{ij} , by an increment δ_{ij} . To be able to compute the time-averaged mass flows \bar{p}_{ijk} and \bar{f}_{ijk} , a mathematical relationship between time-averaged process flows and the batch sizes is required. In logical form the relationship between the feed flowrate and batch size is Eq. 2.

$$\bar{f}_{ijk} = \frac{f_{ijk}}{\tau_{ij}} \text{ if } x_{ij} = 0 \text{ or } \frac{f_{ijk}}{\tau_{ij} + \delta_{ij}} \text{ if } x_{ij} = 1 \quad (2)$$

One way of rewriting Eq. 2 as a mathematical formula is to introduce the binary variable as shown in Eq. 3.

$$\bar{f}_{ijk} = \frac{f_{ijk}}{\tau_{ij} + \delta_{ij} x_{ij}} \quad (3)$$

This form of the relationship, however, is unsatisfactory because f_{ijk} is divided by x_{ij} , violating one of the assumptions stated above. To obtain a satisfactory mathematical relationship, Eq. 3 can be reformulated a different way through the introduction of additional variables. A development is provided in the following subsection.

7.3.2.1. Efficient Formulation of the Design Equations

Equation 3 can be rearranged to give Eq. 4.

$$\bar{f}_{ijk} - \frac{f_{ijk}}{\tau_{ij}} = 0 \quad \text{or} \quad \frac{\delta_{ij} f_{ijk}}{\tau_{ij}(\tau_{ij} + \delta_{ij})}. \quad (4)$$

Eq. 4 has a form commonly encountered in discrete optimization problems; namely, the product of a continuous variable and a discrete variable (Eq. 5).

$$g(u, v) = \tilde{g}(u) \cdot v, \quad (5)$$

where v is a binary variable and $\tilde{g}(u)$ is a positive, bounded, continuous function of a bounded, continuous variable(s), u . It is noted that an exact linearization is needed for Eq. 1 to be used as an explicit constraint in a mathematical programming formulation. Furthermore, the form of the exact linearization can significantly impact computational performance in commonly used algorithms for mixed-integer linear programming and mixed-integer nonlinear programming. It is instructive to consider first the most obvious way to linearize Eq. 5, namely Eqs. 6–8.

$$g(u, v) \leq \tilde{g}(u) \quad (6)$$

$$g(u, v) \leq \tilde{g}^{UP} \cdot v \quad (7)$$

$$g(u, v) \geq \tilde{g}(u) - \tilde{g}^{UP} \cdot (1 - v), \quad (8)$$

where \tilde{g}^{UP} is an upper bound on $\tilde{g}(u)$, *i.e.*, $\tilde{g}(u) \leq \tilde{g}^{UP}$ over the domain of u . The validity of Eqs. 8–10 is intuitive—simply consider the cases $v = 0$ and $v = 1$ separately. The problem with Eqs. 8–10 is that they can give rise to a large relaxation gap, particularly when the value of \tilde{g}^{UP} is not close to the value at the optimum of the mathematical optimization. The nature of the difficulty is shown in Figure 7.8.

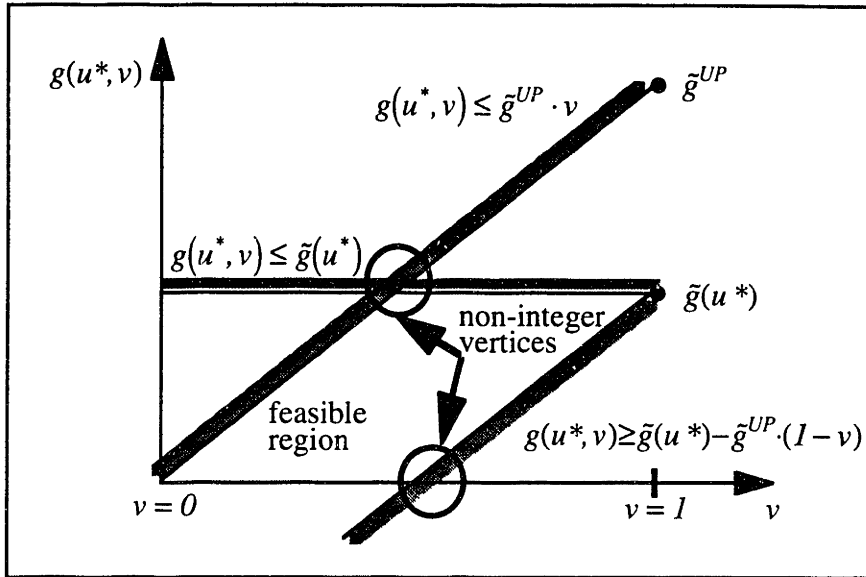


Figure 7.8. Illustration of How Non-integer Vertices Arise Due to Gap Between Upper Bound and True Variable Value. The u^* corresponds to a specific value of u .

Figure 7.8 is a projection of the entire variable space onto the subspace defined by $g(u^*, v)$ and v (u^* corresponds to a specific value of u), ignoring constraints other than Eqs. 6–8. The significance of the non-integer points shown in Figure 7.8 is that the relaxation of the binary variable can take one far away from a feasible solution.

An alternate linearization of Eq. 5 should seek to replace the feasible region shown in Figure 7.8 with a new one where the integrality gap is reduced. Among the techniques available for deriving tighter (*i.e.*, smaller integrality gap) constraints are lifting techniques, wherein additional variables are introduced to produce a formulation in a higher dimensional space that is nevertheless tighter. Thus, in lifting techniques there is a tradeoff between cost of larger problem size versus the benefit of a reduced integrality gap.

For the current problem, it is possible to do lifting by exploiting the fact that g and \tilde{g} are linearly related for fixed u . Thus, the discretization of \tilde{g} into piece-wise linear segments and subsequent equating to g in fact constitute an exact linearization. The first part of the procedure is accomplished by Eqs. 9–15, which discretize the domain of \tilde{g} into $1, \dots, q, q+1, \dots, Q$ segments. The argument u is suppressed to emphasize that the discretization does not involve u .

$$\tilde{g} = \sum_{q=1}^Q \varphi_q \lambda_q \tag{9}$$

$$\lambda_q + \lambda_{q+1} \geq w_q, \quad q = 1, \dots, Q-1 \tag{10}$$

$$\sum_{q=1}^Q \lambda_q = 1 \quad (11)$$

$$\sum_{q=1}^{Q-1} w_q = 1 \quad (12)$$

$$\lambda_q \in [0, 1], \quad q = 1, \dots, Q \quad (13)$$

$$w_q \in \{0, 1\}, \quad q = 1, \dots, Q-1 \quad (14)$$

$$\varphi_1 = 0 < \varphi_2 < \dots < \varphi_{Q-1} < \varphi_Q = \bar{g}^{UP} \quad (15)$$

In Eqs. 9–15, the φ_q are fixed constants that discretize the domain of \bar{g} . Interpolation between φ_q and φ_{q+1} , $q \in \{1, \dots, Q-1\}$, is accomplished with the continuous variables λ_q , $q \in \{1, \dots, Q\}$. The remaining variables and constraints are required to specify a unique choice for the λ_q . Of particular importance are the binary variables w_q , $q \in \{1, \dots, Q-1\}$, which are responsible for uniquely (Eq. 12) assigning the λ_q .

The second part of the procedure consists of multiplying Eq. 9 by v , resulting in Eqs. 16 and 17.

$$\bar{g} = \sum_{q=1}^Q \varphi_q z_q \quad (16)$$

$$z_q = \lambda_q \cdot v, \quad q = 1, \dots, Q \quad (17)$$

Because the λ_q are constrained to the unit interval and v is binary, the z_q are constrained to the unit interval. It follows that the z_q are exactly linearizable by Eqs. 18–20.

$$z_q \leq v, \quad (18)$$

$$z_q \leq \lambda_q, \text{ and} \quad (19)$$

$$z_q \geq v + \lambda_q - 1, \quad q = 1, \dots, Q. \quad (20)$$

Equations 18–20 are collectively known as the Adams-Sherali polytope, which contains only integral vertices. The increased tightness of Eqs. 18–20 over Eqs. 6–8 is established by the following proof.

Theorem 1. Eqs. 18–20 imply Eqs. 6–8; the implication is strict if $z_N = 0$.

Proof of Theorem 1:

The basis for comparison is derived by substituting Eqs. 9 and 16 into Eqs. 6–8, resulting in Eqs. 21–23.

$$\sum_{q=1}^Q \varphi_q z_q \leq \sum_{q=1}^Q \varphi_q \lambda_q \tag{21}$$

$$\sum_{q=1}^Q \varphi_q z_q \leq \varphi_N \cdot v \tag{22}$$

$$\sum_{q=1}^Q \varphi_q z_q \geq \sum_{q=1}^Q \varphi_q \lambda_q - c_N \cdot (1 - v) \tag{23}$$

By inspection, Eq. 19 \Rightarrow 21. Next, Eq. 22 is rearranged by dividing through by φ_N (Eqs. 24–25).

$$\sum_{q=1}^Q \varpi_q z_q \leq v \tag{24}$$

where

$$\varpi_q \equiv \frac{\varphi_q}{\varphi_N}. \tag{25}$$

It follows immediately from Eq. 25 that $\varpi_q < 1$ if $q < Q$; $\varpi_Q = 1$. In particular,

$$\sum_{q=1}^Q \varpi_q z_q \leq \sum_{q=1}^Q z_q. \tag{26}$$

Furthermore, the inequality is strict if $z_q = 0$, $q < Q$. It follows from Eq. 18 that

$\sum_{q=1}^Q z_q \leq v$; thus Eq. 18 \Rightarrow 22:

$$\sum_{q=1}^Q \varpi_q z_q < \sum_{q=1}^Q z_q \leq v \text{ if } z_N = 0 \tag{27}$$

$$\sum_{q=1}^Q \varpi_q z_q \leq \sum_{q=1}^Q z_q \leq \nu \text{ if } z_N > 0. \quad (28)$$

Finally, rearrangement of Eqs. 23 and 20 yields Eq. 29 and 30.

$$\sum_{q=1}^Q \varpi_q (\lambda_q - z_q) \leq 1 - \nu \quad (29)$$

$$\sum_{q=1}^Q \lambda_q - z_q \leq 1 - \nu \quad (30)$$

By the same reasoning as in the last case it follows that

$$\sum_{q=1}^Q \varpi_q (\lambda_q - z_q) < \sum_{q=1}^Q \lambda_q - z_q \leq 1 - \nu \text{ if } z_N = 0 \quad (31)$$

$$\sum_{q=1}^Q \varpi_q (\lambda_q - z_q) \leq \sum_{q=1}^Q \lambda_q - z_q \leq 1 - \nu \text{ if } z_N > 0. \quad (32)$$

Thus, Eq 20 \Rightarrow 23. ///

To summarize, the tightened exact linearization of Eq. 2 consists of Eqs. 9–16 and 18–20. It should be noted that the absolute relaxation gap generated by this formulation becomes smaller with increasing Q . The reduced gap is offset, however, by the increased dimensionality of the lifted space. Thus, the best choice of Q is that which minimizes the overall computational effort. Such a choice is best determined through computational experimentation.

The lifting strategy can be employed to derive an efficient representation of Eq. 2 by adapting the variables introduced in the derivation to the problem at hand. In particular, the batch size of a feed, f_{ijk} , can be discretized by making the following assignments:

z_q becomes z_{ijkq}

λ_q becomes λ_{ijkq}

φ_q becomes φ_{ijkq}

w_q becomes w_{ijkq}

ν becomes x_{ij}

where

$$q = 1, \dots, Q, (i, j) \in I \times J, k \in K_0(i, j).$$

The resulting discretization of f_{ijk} is given by Eqs. 33–36.

$$f_{ijk} = \sum_{q=1}^Q \lambda_{ijkq} \varphi_{ijkq}, \quad (i, j) \in I \times J, k \in K_0(i, j) \quad (33)$$

$$\lambda_{ijkq} + \lambda_{ijk(q+1)} \geq w_{ijkq}, \quad q = 1, \dots, Q-1 \quad (34)$$

$$\sum_{q=1}^Q \lambda_{ijkq} = y_{ij} \quad (35)$$

$$\sum_{q=1}^{Q-1} w_{ijkq} = y_{ij} \quad (36)$$

- $K_0(i, j)$ = feed stream-type to unit process (i, j) .
 λ_{ijkq} ($\in [0, 1]$) = variables specifying linear combinations of the φ_{ijkq} .
 φ_{ijkq} ($\in [f_{ijk}^{LO}, f_{ijk}^{UP}]$) = constant discrete values of the batch size.
 w_{ijkq} ($\in \{0, 1\}$) = variables specifying the piece-wise linear segment.
 y_{ij} ($\in \{0, 1\}$) = variable denoting existence of the unit process.

Incorporating Eqs. 16 and 33–36 into Eq. 4, the relationship between the batch charge size and time-averaged flowrate is formulated as Eq. 37.

$$\bar{f}_{ijk} = \frac{f_{ijk}}{\tau_{ij}} - \frac{\delta_{ij}}{\tau_{ij}(\tau_{ij} + \delta_{ij})} \sum_{q=1}^Q z_{ijkq} \varphi_{ijkq}, \quad (i, j) \in I \times J, k \in K_0(i, j) \quad (37)$$

z_{ijkq} ($\in [0, 1]$) = variable used in exact linearization of the product $\lambda_{ijkq} x_{ij}$.

A set of discretized flow equations are also written for the time-averaged flowrates, \bar{f}_{ijk} , with analogous terms: \bar{z}_{ijkq} , $\bar{\lambda}_{ijkq}$, $\bar{\varphi}_{ijkq}$, \bar{w}_{ijkq} , and \bar{x}_{ij} .

7.3.3. Material Conversion Design Equations

Material conversion design equations consist of mass balances between the feed flow entering a unit process and the set of products leaving the unit process. The key set of parameters in these equations are material conversion factors, θ_{ijm} , where m is one of the several stream-types that can be produced by a unit process. These parameters are obtained directly from simulation of chemical conversions and separations in the base case designs

used to parameterize the unit process. In the simple model treated here, the discrete decision of discarding a by-product isomer as waste ($x_{ij} = 0$), versus recovering a second product to yield a reduced waste stream ($x_{ij} = 1$), is modeled by a set of mass balances (Eq. 38) and constraints (Eqs. 39 and 40) that incorporate the material conversions factors.

$$\bar{p}_{ijm} = \theta_{ijm} \bar{f}_{ijk}, \quad (i, j) \in I \times J, \quad k \in K_0(i, j), \quad m \in K_1(i, j) \quad (38)$$

$$\bar{p}_{ijs} \leq \theta_{ijs} \sum_{q=1}^Q \bar{z}_{ijkq}^1 \bar{\varphi}_{ijkq}, \quad s \in K_2(i, j) \cup K_3(i, j) \quad (39)$$

$$\bar{p}_{ijs} \geq \theta_{ijs} \sum_{q=1}^Q (\bar{\lambda}_{ijkq} - \bar{z}_{ijkq}^2) \bar{\varphi}_{ijkq} \quad (40)$$

- $K_1(i, j)$ = primary product stream-type to unit process (i, j) .
 $K_2(i, j)$ = optional product stream-type ($x_{ij} = 1$) to unit process (i, j) .
 $K_3(i, j)$ = waste stream-type [if $x_{ij} = 1$] to unit process (i, j) .
 $K_4(i, j)$ = waste stream-type [if $x_{ij} = 0$] to unit process (i, j) . (Not shown)
 $\bar{\varphi}_{ijkq}$ ($\in [\bar{f}_{ijk}^{LO}, \bar{f}_{ijk}^{UP}]$) = constant discrete values of the time-averaged flow.
 \bar{z}_{ijkq}^1 ($\in [0, 1]$) = variable used in exact linearization of the product $\bar{\lambda}_{ijkq} x_{ij}$.
 $\bar{\lambda}_{ijkq}$ ($\in [0, 1]$) = variables specifying linear combinations of the $\bar{\varphi}_{ijkq}$.
 \bar{z}_{ijkq}^2 ($\in [0, 1]$) = variable used in exact linearization of the product $\bar{\lambda}_{ijkq} (1 - x_{ij})$.

Since the terminal waste streams, $k \in K_3(i, j) \cup K_4(i, j)$, are not fed back into the process, their conversion factors are set to unity and their costs made directly proportional to the feed size.

7.3.4. Unit Process Specifications

A key component of unit process specification is the estimation of equipment and utility requirements. As discussed in the introduction, the unit operations and utilities are related to base case design estimates according to scaling factors, which multiply the capacities (*e.g.*, reactor volumes, heat exchanger surface areas, *etc.*) of a base case design to the required level. Formally, each unit process $(i, j) \in I \times J$ has a set of unit operations,

$M(i, j)$, whose capacities are defined by factors, s_{ijm} , $m \in M(i, j)$. Similarly, utilities are members of a set $R(i, j)$ with scaling factors \bar{s}_{ijr} , $r \in R(i, j)$.

"Power rule" correlations are employed to estimate equipment costs as a function of the scaling factors (Peters and Timmerhaus, 1980). The capital cost, c_{ijm} , of each unit operation $m \in M(i, j)$ is estimated by an equation of the form

$$c_{ijm} = \alpha_{ijm}^0 \cdot (s_{ijm})^{\beta_{ijm}^0} + \chi_{ijm}^0 y_{ij}, \quad m \in M(i, j), \quad (i, j) \in I \times J, \quad (41)$$

where the parameters α_{ijm}^0 , β_{ijm}^0 , and χ_{ijm}^0 are statistically fitted to quoted costs. Similarly, the net present value \bar{c}_{ijr} of each utility $r \in R(i, j)$ is given by

$$\bar{c}_{ijr} = \bar{\alpha}_{ijr}^0 \cdot (\bar{s}_{ijr})^{\bar{\beta}_{ijr}^0} + \bar{\chi}_{ijr}^0 y_{ij}, \quad r \in R(i, j), \quad (i, j) \in I \times J. \quad (42)$$

In light of the large number of possible scaling factors, which could lead to computationally difficult problems, Eq. 42 is simplified by assuming that the set of s_{ijm} can be related directly to the batch size of the feed. The relationship takes the form of Eq. 43 so as to preserve the functional form established in Eq. 41.

$$s_{ijm} = \gamma_{ijm} \cdot (f_{ijk})^{K_{ijm}}, \quad m \in M(i, j), \quad k \in K_0(i, j), \quad (i, j) \in I \times J. \quad (43)$$

Similarly, it is assumed that the set of \bar{s}_{ijr} can be scaled directly to the time averaged flowrate of the feed (Eq. 44).

$$\bar{s}_{ijr} = \bar{\gamma}_{ijr} \cdot (\bar{f}_{ijk})^{\bar{K}_{ijr}}, \quad r \in R(i, j), \quad k \in K_0(i, j), \quad (i, j) \in I \times J. \quad (44)$$

Inserting Eqs. 43 and 44 into Eqs. 41 and 42 yields the general model form (Eqs. 45 and 46).

$$c_{ijm} = \alpha_{ijm} \cdot (f_{ijk})^{\beta_{ijm}} + \chi_{ijm} y_{ij}, \quad m \in M(i, j), \quad k \in K_0(i, j), \quad (i, j) \in I \times J \quad (45)$$

$$\bar{c}_{ijr} = \bar{\alpha}_{ijr} \cdot (\bar{f}_{ijk})^{\bar{\beta}_{ijr}} + \bar{\chi}_{ijr} y_{ij}, \quad r \in R(i, j), \quad k \in K_0(i, j), \quad (i, j) \in I \times J \quad (46)$$

where

$$\alpha_{ijm} \equiv \gamma_{ijm} \alpha_{ijm}^0 \quad (47)$$

$$\bar{\alpha}_{ijr} \equiv \bar{\gamma}_{ijr} \bar{\alpha}_{ijr}^0 \quad (48)$$

$$\beta_{ijm} \equiv \kappa_{ijm} \beta_{ijm}^0 \quad (49)$$

$$\bar{\beta}_{ijr} \equiv \bar{\kappa}_{ijr} \bar{\beta}_{ijr}^0 \quad (50)$$

$$\chi_{ijm} \equiv \chi_{ijm}^0 \quad (51)$$

$$\bar{\chi}_{ijr} \equiv \bar{\chi}_{ijr}^0 \quad (52)$$

7.3.5. Mixed-Integer Linear Reformulation

Due to the nonconvex equations introduced in the previous subsection, the optimization problem is a nonconvex mixed-integer nonlinear program. Exploiting the separability of the equipment and utility cost equations, discretization of the mass flows (*e.g.*, Eqs. 33-36) allows the nonconvex functions to be linearized into piece-wise linear segments. This technique is known as the λ -technique and allows programming problems with separable nonlinearities to be approximated as mixed-integer linear programs. The form of the approximation is given in Eqs. 53 and 54.

$$\alpha_{ijm} \cdot (f_{ijk})^{\beta_{ijm}} = \alpha_{ijm} \cdot \left(\sum_{q=1}^Q \varphi_{ijkq} \lambda_{ijkq} \right)^{\beta_{ijm}} \approx \alpha_{ijm} \cdot \sum_{q=1}^Q (\varphi_{ijkq})^{\beta_{ijm}} \lambda_{ijkq} \quad (53)$$

$$\bar{\alpha}_{ijr} \cdot (\bar{f}_{ijk})^{\bar{\beta}_{ijr}} = \bar{\alpha}_{ijr} \cdot \left(\sum_{q=1}^Q \bar{\varphi}_{ijkq} \bar{\lambda}_{ijkq} \right)^{\bar{\beta}_{ijr}} \approx \bar{\alpha}_{ijr} \cdot \sum_{q=1}^Q (\bar{\varphi}_{ijkq})^{\bar{\beta}_{ijr}} \bar{\lambda}_{ijkq} \quad (54)$$

where

$$m \in M(i, j), r \in R(i, j), k \in K_0(i, j), (i, j) \in I \times J.$$

Substituting Eqs. 53 and 54 into Eqs. 45 and 46 yields the desired relations (Eqs. 55 and 56).

$$c_{ijm} = \alpha_{ijm} \cdot \sum_{q=1}^Q (\varphi_{ijkq})^{\beta_{ijm}} \lambda_{ijkq} + \chi_{ijm} y_{ij} \quad (55)$$

$$\bar{c}_{ijr} = \bar{\alpha}_{ijr} \cdot \sum_{q=1}^Q (\bar{\varphi}_{ijkq})^{\bar{\beta}_{ijr}} \bar{\lambda}_{ijkq} + \bar{\chi}_{ijr} y_{ij} \quad (56)$$

7.3.6. Additional Mathematical Constraints

Several additional constraints can be derived to strengthen the mathematical programming formulation. Recalling the discretization technique used to derive an exact linearization between f_{ijk} and \bar{f}_{ijk} , it should be noted that an additional constraint can be derived by combining the λ_{ijkq} and w_{ijkq} (or $\bar{\lambda}_{ijkq}$ and \bar{w}_{ijkq}) directly. Equation 57 provides an illustration.

$$\sum_{q=1}^{q^*-1} \lambda_{ijkq} + w_{ijkq^*} + \sum_{q=q^*+2}^Q \lambda_{ijkq} \leq y_{ij}, \quad q^* \in \{1, \dots, Q-1\}, \quad (i, j) \in I \times J \quad (57)$$

In the case where $q^* = 1$ the first summation is omitted. Similarly, the second summation is omitted if $q^* = Q - 1$. Equation 57 expresses the fact that $w_{ijkq^*} = 1$ forces $\lambda_{ijkq} = 0, q = 1, \dots, (q^* - 1), (q^* + 2), \dots, Q$.

A second set of constraints can be defined from specific structural features of the reaction path network. For example, identical parallel unit processes (*e.g.*, two chlorination processes that use the same feed and produce the same product) introduce degenerate solutions into the mathematical programming formulation. Such degeneracy can considerably increase the computational effort required to prove optimality of a solution. The degeneracy can be removed by adding constraints of the form $y_{ij} \leq y_{i^*j^*}$, which require that one unit be arbitrarily chosen over the second.

7.4. Model Extensions

In principle, the mathematical programming formulation can be extended in several directions, including simultaneous scheduling of unit-to-task assignments, sensitivity analysis of unit process model parameters, and optimization of variable operating conditions and split fractions. A key practical consideration in extending the model is the presence of uncertainty in the formulation's constraints and objective function. If rigorous design tools are used to obtain model parameters, the introduction of new information and new design concepts over time can still create uncertainty in many aspects of the formulation. At the level of unit process configurations, the separation sequences, reactor design specifications, and cycle times may be modified to reflect improved thermophysical measurements. At the total plant level, production targets, waste stream costs, and utility costs can vary considerably to create stochastic uncertainty in time-varying model parameters.

The presence of diverse sources of uncertainty motivate explicit treatments in the mathematical programming formulation. Parametric analysis techniques, for example, can be employed to identify critical model parameters which affect feasibility and robustness of proposed designs (an overview of relevant techniques is presented in appendix three). Computational results obtained can be used to establish priorities for further analysis to reduce the variance in model parameters. The analysis might include, for example, development of improved models or new experimental measurements.

A second class of techniques are stochastic optimization techniques, which incorporate uncertainty analysis directly in the solution of the mathematical programming formulation (a brief discussion is presented in appendix three). Uncertainty in future production targets, raw material costs, and other factors can significantly influence long term profitability. Stochastic optimization techniques allow "robust" designs to be selected to minimize the impact of deviations from nominal design assumptions. A technique widely used in stochastic optimization is chance-constrained programming, wherein a set of constraints in the mathematical program are satisfied in a probabilistic sense. In effect, the slack variables in the constraints are treated as random variables. The formulation yields tunable parameters that can be used to study the tradeoff between the objective function and the probability of satisfying the constraints containing the uncertainty. In other words, conservative designs with "modest" profitability can be chosen which almost certainly satisfy the constraints, or bolder, more profitable designs can be chosen that are more likely to result in violation of the constraints.

In the next section, a probabilistic treatment of uncertain production targets is considered employing a chance-constrained programming strategy. Given the discrete joint probability distribution describing the production objectives, the mathematical program is optimized using an expected value criterion.

7.4.1. Probabilistic Treatment of Uncertain Production Targets

Uncertainty in production targets can arise for a number of reasons, including the introduction of new products, changes in product regulation, and changes in market preferences. Ideally, the initial process configuration should be chosen to maximize the net present value of the process given the possibility that one or more of the products will be eliminated in a time frame that is short compared to the lifetime of the project or campaign. To incorporate this goal into the mathematical programming formulation, the mathematical programming strategy is extended to include the choice among products in the presence of discrete uncertainty in production targets.

For each product stream-type, $k \in K_p$, a binary variable, π_k , is introduced where $\pi_k = 1$ if the product is produced and $\pi_k = 0$ otherwise. One would obviously like to constrain the set of products to avoid those that are ultimately to be eliminated from consideration. This objective can be stated mathematically as Eq. 58.

$$\bar{P}_k \leq \bar{P}_k^{UP} \pi_k, \quad k \in K_p, \quad (58)$$

where \bar{P}_k^{UP} is given by the production quota for the stream-type. The realization of π_k , however, may not be known with certainty. As a result, the constraint (Eq. 58) may only be satisfied probabilistically.

A variable such as π_k is known as a random variable. It should be noted that, in general, subsets of π_k can be correlated random variables. Therefore, it is useful to describe the random variables jointly as a vector $\pi = \{\pi_{k_t}\} = (\pi_1, \pi_2, \dots, \pi_m)$, where $k_t \in K_p$ is denoted by t to simplify the notation. Each of the 2^m possible realizations of the vector π have an associated probability, ρ_e , where e is an index to the outcome of π . The set of indices to the entire set of 2^m possible realizations is denoted by E , which forms a probability space. The set of probabilities ρ_e , $e \in E$, must add to unity (Eq. 59).

$$\sum_{e \in E} \rho_e = 1 \quad (59)$$

To obtain a probabilistic equivalent to Eq. 58, it is useful to rearrange the equation to isolate the random variable π_t .

$$\pi_t \geq \tilde{p}_t, \quad k_t \in K_p, \quad (60)$$

where

$$\tilde{p}_t \equiv \frac{\bar{P}_{k_t}}{\bar{P}_k^{UP}}. \quad (61)$$

Letting $T^m \equiv \{1, \dots, m\}$ be the set of product stream-type indices, the probabilistic form of Eq. 58 for the correlated set may be written as a chance-constraint (Eq. 62).

$$P[\pi_1 \geq \tilde{p}_1 \text{ and } \dots \text{ and } \pi_t \geq \tilde{p}_t \text{ and } \dots \text{ and } \pi_m \geq \tilde{p}_m] \geq a, \quad a \in [0, 1] \quad (62)$$

The new variable a is a decision variable which represents the probability of the constraint being violated. An a value near unity means that little risk will be taken in selecting products. As in the case of the π_t , the variables \tilde{p}_t can be described as $\tilde{p} = (\tilde{p}_1, \tilde{p}_2, \dots, \tilde{p}_m)$. Incorporating the definition for \tilde{p}_t , Eq. 62 can be rewritten as shown in Eq. 63.

$$P[\pi \geq \bar{p}] \geq \alpha \quad (63)$$

The conversion of the chance-constraint given by Eq. 63 into deterministic constraints is accomplished by recognizing that there are only a finite number of outcomes for $P[\pi \geq \bar{p}]$ corresponding to the 2^m possible realizations of π . Since $\bar{p} \in [0, 1]^m$, it is apparent that

$$P[\pi^e \geq \bar{p}] = \rho_e, \quad e \in E, \quad (64)$$

where π^e denotes a realization of π . Based on Eq. 64 the following theorem is readily established.

Theorem 2. The following deterministic constraints are equivalent to Eq. 63:

$$\sum_{t \in D_e^+} \tilde{\pi}_t - \sum_{t \in D_e^-} \tilde{\pi}_t \leq |E_e^+| - 1 + \mu_e, \quad e \in E \quad (65)$$

$$\alpha \leq (1 - \mu_e) + \rho_e, \quad e \in E, \text{ and} \quad (66)$$

$$\sum_{e \in E} \mu_e = 1, \quad (67)$$

where

$$\tilde{\pi}_t = \begin{cases} 1 & \text{if } \bar{p}_t > 0 \\ 0 & \text{if } \bar{p}_t = 0, \end{cases} \quad (68)$$

$$D_e^+ \equiv \{t \mid \tilde{\pi}_t = 1 \text{ in the particular outcome } e \in E\} \quad (69)$$

$$D_e^- \equiv \{t \mid \tilde{\pi}_t = 0 \text{ in the particular outcome } e \in E\}. \quad (70)$$

Proof of Theorem 2:

First, Eq. 71 shows that any realization e for which $\rho_e < \alpha$ is not a feasible solution to Eqs. 65–70.

$$\begin{aligned} \rho_e < \alpha &\Rightarrow \mu_e = 0 \Rightarrow \sum_{t \in D_e^+} \tilde{\pi}_t - \sum_{t \in D_e^-} \tilde{\pi}_t \leq |E_e^+| - 1 \\ &\Rightarrow \exists t^* \in D_e^- \text{ such that } \tilde{\pi}_{t^*} = 1 \end{aligned} \quad (71)$$

Conversely, any realization e satisfying $\rho_e > \alpha$ is a valid solution to Eqs. 65–66, 68–70. Equation 67 selects one of the valid solutions by setting $\mu_{e^*} = 1$, $e^* \in E$. Eq. 71 shows that all other solutions will be excluded due to $\mu_e = 0$, $e \neq e^*$.////

7.5. Choice of Objective Functions

A mixed-integer linear programming strategy is capable of incorporating several types of objective functions. A general template for the construction of a useful class of objective functions is given in Eq. 72.

$$Z = \vartheta \cdot \Phi[v(\omega^*)] + (1 - \vartheta) \cdot \int_{\Omega} \Phi[v(\omega)] d\omega + L's + (1 - \vartheta) \cdot T' \alpha_{\Omega} \quad (72)$$

The function $\Phi[\bullet]$ is a measure of the net present value of the chemical process, as a function of ω^* , a vector of nominal values for net material and energy costs and equipment costs over the production time horizon. To deal with random variables that are not time-dependent, the second term in Eq. 72 gives the net present value of the economic function over a scalar probability space Ω . The term $L's$ is a linear penalty term consisting of the scalar product of a weighting vector L and selected positive variables, s (e.g., waste stream flowrates). The term $T' \alpha$ is a second linear penalty term that incorporates the vector of chance-constrained variables, α_{Ω} , corresponding to the probability space, Ω . The term T can be interpreted as a weighting of the importance of satisfying the penalized constraints; a large positive weighting reduces the contribution of $\Phi[\bullet]$ to the objective function. Increasingly large values of T favor larger α_{Ω} , causing a more conservative design to be chosen. The parameter $\vartheta \in [0, 1]$ is a linear weighting parameter that controls the relative proportions of deterministic and probabilistic contributions in the objective function.

Variation of the parameters L , T , and ϑ allow construction of tradeoff curves such as depicted in Figure 7.9. Alternate designs obtained from the tradeoff curves tend to exhibit different robustness properties. In the neighborhood of optimal design there are often different configurations that are much more robust to changes in parameter values while making only a modest compromise in the economic profit.

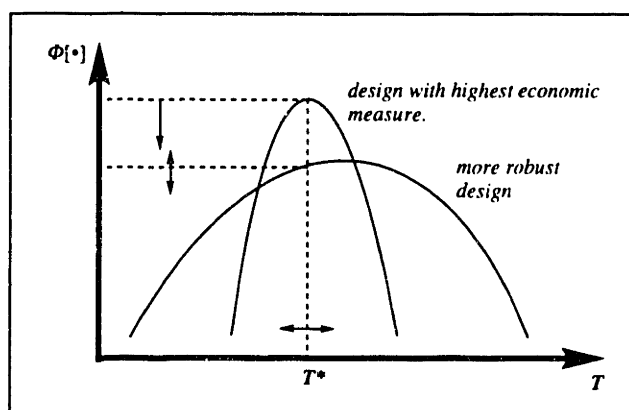


Figure 7.9. Comparison of Optimal and Robust Unit Process Configurations.

7.6. Computer Implementation

The mathematical programming formulations developed in this chapter are implemented in the GAMS (General Algebraic Modeling System), Version 2.25 (Brooke, *et al.*, 1988) and solved with the (Optimization Subroutine Library) Version 2.001. Several computational platforms have been employed, including a Digital VAX 6320 and a Hewlette-Packard 9000/735.

7.7. Conclusions

Synthesis of integrated unit process configurations has been proposed for analysis of tradeoffs in the choice of chemical reaction paths producing related components. The appropriate economy-of-scale for the method is based on the reaction path network concept, which contains strategic raw materials, intermediate, and chemical conversions to a set of products. Information assumptions reasonable for the early stages of process development allow an efficient mixed-integer linear programming formulation to be developed. Extension of the mathematical programming formulation potentially allows results to be used in experimental design, sensitivity analysis, flexibility analysis, and forecasting. A chance-constrained augmentation of the formulation has been derived to allow incorporation of uncertainty in future production targets in the choice of the optimal unit process configuration. Computational examples involving large-scale reaction path networks are essential for verification of both the deterministic and probabilistic formulations (chapter ten).

Chapter 8

Verification of Reaction Path Synthesis Algorithm on a Large Case Study Involving a Spectrum of Related Products

8.1. Introduction

Chapter two has introduced a reaction path synthesis strategy for the generation and analysis of reaction path networks. Manufacturing strategies incorporating reaction path networks offer several inherent advantages, including

- increased utilization of raw materials through the selection of ubiquitous raw materials;
- reduced waste streams through optimization of a sequence of compound refunctionalizations; and
- reduction in the number of process tasks.

Balancing these benefits, tradeoffs to be considered include

- reduced process flexibility due to interdependencies between yields and selectivities in reaction paths;
- more difficult separations for the recovery of multiple reaction products; and
- possible impurity reactions, process technology implications, waste streams, and so on.

A class of chemistry that exhibits both the benefits and tradeoffs listed above consists of diazotization of the amino compounds and the various conversions that can be applied to the aromatic diazonium ions (*i.e.*, $\text{Ar}-\text{N}\equiv\text{N}^+$, where Ar denotes the aromatic substrate). Illustrative conversions are shown in Figure 8.1.

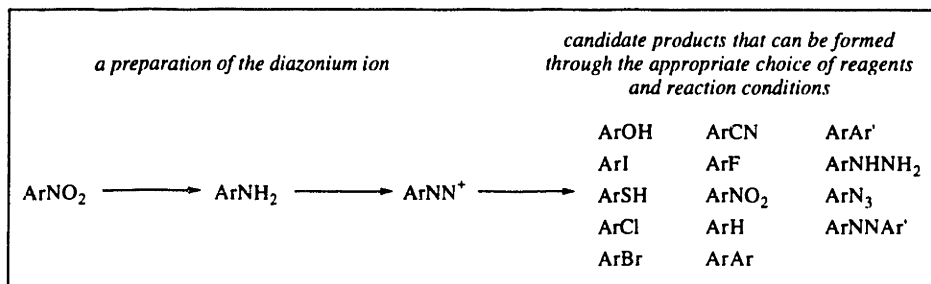


Figure 8.1. Common Diazotization Sequences used in the Preparation of Specialty and Fine Aromatic Compounds.

Such chemistry is well-studied and used often in the synthesis of dye intermediates and other specialty and fine chemicals. The development of each application, however, is non-trivial due to a rich set of possible competing reactions, safety considerations, and environmental constraints. Thus, while the chemistry of the diazo group can be well-suited to reaction networks when combined with other chemical conversions (*e.g.*, nitration and amination as illustrated in Figure 8.1), significant tradeoffs are present that might make a particular conversion either very desirable or perhaps economically and environmentally unacceptable. As such, diazo chemistry is a good basis for development and verification of screening criteria designed to take a large number of candidate routes and reduce them to a manageable number.

This chapter presents a case study in which diazo chemistry plays a key role in the synthesis of a series of five substituted benzonitrile compounds. These compounds (Figure 8.2) are used as intermediates to dyes and other specialty chemicals.

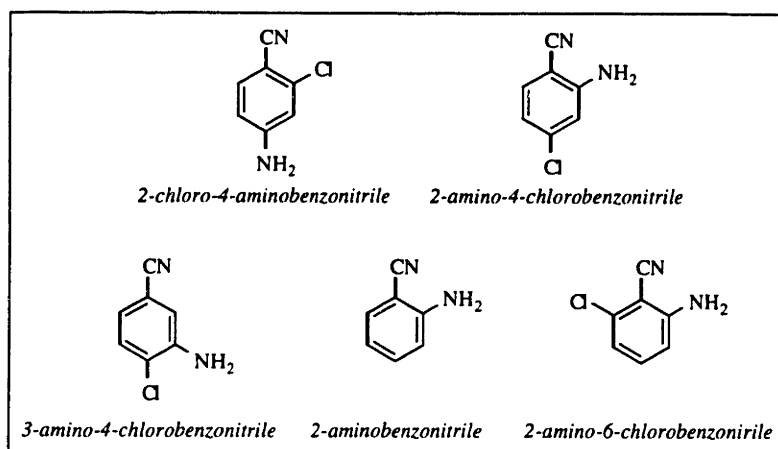


Figure 8.2. Series of Benzonitrile Dye Intermediates.

While several routes are available to these compounds, economic scale of production and cyclical market demand probably favor manufacture within a single multipurpose batch plant. The generation of candidate reaction path networks should provide both alternatives for multipurpose production as well as a head-to-head comparison with existing industrial routes. The specific objectives of the case study presented are as follows:

- demonstrate that the retrosynthetic analysis approach can generate both the existing routes as well as alternate routes and raw material sources;
- evaluate the utility of the reaction prediction techniques described in the last chapter in reducing an extremely large number of alternate paths to a relatively small number of promising routes;
- characterize missing information required to improve reaction prediction; and
- identify tradeoffs that must be considered in a preliminary process evaluation.

The chapter begins with a definition of the chemical reactions and the reaction prediction techniques to be employed. Next, two types of computational results are presented. First, the number of paths generated for different choices of screening criteria (*e.g.*, maximum number of chemical conversions in a reaction path; lower bounds on reaction rate and selectivity) are considered. Second, the reaction path network is examined. The potential industrial use of the reaction network and additional information requirements are considered in turn.

8.2. Overview

A large-scale case study for the synthesis of reaction networks is performed for a set of five benzonitrile dye intermediates. Available unit conversions include nitration, chlorination, the Béchamp reduction, ammonolysis, hydro-dediazotation, and the Sandmeyer reaction. Reaction rules and quantitative screening criteria based on a linear free-energy relationship are employed to systematically evaluate chemical conversions. The number of paths generated is shown to depend most strongly on the hydro-dediazotation transform and the reaction rules. Within the reaction path network, a number of existing routes based on ammonolysis of dichloro-nitrobenzenes are correctly identified. In addition, less selective nitrations and chlorinations yielding usable by-products are identified. The latter allow a new set of strategic raw materials to be proposed, namely nitro- and chloroanilines. Whereas the ammonolysis reactions are performed in a second process and are increasingly subject to environmental regulation, the routes based on nitrated and chlorinated anilines can potentially be folded into the existing process. In addition to these routes, the use of the hydro-dediazotation transform yields even more highly integrated, though probably not industrially practical, synthesis routes involving tetrasubstituted intermediates. The potential role of simplifying transforms such as hydro-dediazotation is discussed. Also likely impurities and wastes are assessed to provide a basis for preliminary process screening. The computational code and production runs for the benzonitrile case study are given in appendix six.

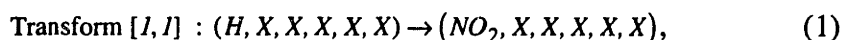
8.3. Chemical Conversions

Because of its widespread use in the dye and pigment industries, the Sandmeyer reaction sequence, which consists of diazotization to convert an amino group to a diazonium ion, followed by cyanation, is considered for the introduction of a cyano group (Hagedorn, 1985). Other reactions considered include homogeneous nitration (Paul, 1958); chlorination (MacMullin, 1948; Weigandt *et al.*, 1951); the Béchamp reduction, which is the conversion of a nitro group to an amino group (Werner *et al.*, 1958); and ammonolysis (Baudet, 1924). Finally, diazotization and hydro-de-diazonation, together comprising the replacement of a diazonium ion by hydrogen (Hagedorn, 1985), are included to permit the removal of an amino group.

The following subsections provide details on the modeling of these chemical conversions. In order to present the modeling issues clearly and concisely, numerous details about the intrinsic process chemistry are saved for chapter ten, which deals with preliminary process screening.

8.3.1. Homogeneous Nitration (Monosubstitution Transform Network)

Nitration of aromatic compounds is the replacement of a hydrogen atom by a nitro group *via* electrophilic aromatic substitution. Using the notation introduced in the previous chapter, the associated transform is given in Eq. 1.



where $[R, T]$ is a chemical transform index in which R denotes the overall reaction and T the specific chemical transform. In Eq. 1, $([I, \bullet])$ is the overall homogeneous nitration reaction and $([\bullet, I])$ the main transformation that replaces hydrogen by the nitro group.

In general, amino substituents must be protected from oxidation during nitration. The most widely used method of protection is to convert the substituted aniline to an acetanilide, followed by nitration in homogeneous media such as acetic anhydride (Streitwieser and Heathcock, 1981). The reactions required to effect the conversion and recovery can be treated separately from the nitration. Rate and selectivity analysis of Eq. 1 is based on the additivity treatment (chapter three) for nitration in acetic anhydride and 2M nitric acid at 25°C. The partial rate factors for the necessary substituents are given in Table 8.1. These parameters have been computed from parameters given by Stock (1968) and by Taylor (1990) (see the supplementary section at the end of the chapter).

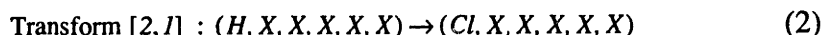
Table 8.1. Nitration Partial Rate Factors for the Estimation of Chemical Reaction Rates.

Substituent	Nitration		
	<i>ortho</i>	<i>meta</i>	<i>para</i>
Cl	0.13	8.4E-4	0.13
CN	9.28E-5	4.36E-4	2.15E-5
NO ₂	2.88E-6	4.17E-5	2.96E-7
NHAc	8921	0.33	5248

Nitration is subject to large deviations from additivity, particularly of polysubstituted substrates containing both strongly activating and deactivating substituents. The predictions are useful, however, for excluding manifestly infeasible reactions, provided the bound on the relative reaction rate bound is set to a low value, *ca.* 10⁻⁵. Multiple substitutions are not considered in light of the strongly deactivating effect of the nitro group (Table 8.1). It is also noted that a diffusion limitation is not considered since the reaction occurs homogeneously in a single phase.

8.3.2. Molecular Chlorination (General Polysubstitution Transform Network)

Chlorination of aromatic compounds is the replacement of a hydrogen atom by a chlorine atom *via* electrophilic aromatic substitution (Eq. 2).



Direct chlorination of an aniline can yield significant polysubstitution. To avoid this problem, the aniline can be converted to an acetanilide as in the case of nitration, the amide group being of lower activating influence. The resulting acetic acid medium is a suitable reaction medium for chlorination by molecular chlorine (Taylor, 1990). Partial rate factors at 25°C are given in Table 8.2. These parameters have been computed from parameters given by Stock (1968) and by Taylor (1990) (see the supplementary section at the end of the chapter).

Table 8.2. Chlorination Partial Rate Factors for the Estimation of Chemical Reaction Rates.

Substituent	Chlorination		
	<i>ortho</i>	<i>meta</i>	<i>para</i>
Cl	0.019	6.54E-4	0.0654
CN	5.38E-7	1.71E-6	1.35E-7
NO ₂	6.28E-9	3.03E-8	1.12E-9
NHAc	6.1E5	0.15	2.5E6

Analytic estimation of the yield loss due to multiple substitution has been illustrated for the chlorination of phenols (chapter two). In the present situation it is possible to exploit the problem structure to simplify the reaction prediction problem. First, one may reasonably assume that economical raw materials will be at least disubstituted compounds, since large-scale and efficient processes already exist for the manufacture of these compounds (*e.g.*, nitroanilines, chloroanilines, *etc.*). Second, Table 8.2 indicates that the cyano and nitro groups are so strongly deactivating (partial rate factors *ca.* 10⁻⁶–10⁻⁹) that any feasible chlorinations will be carried out only on those substrates which already contain an amide group. Inspection of target molecules reveals that two types of raw materials satisfy the simplifying assumptions: *ortho* and *para* substituted compounds in which one of the groups is an amide. For such raw materials, the relative magnitudes of partial rate factors indicate that reaction products should be almost exclusively *ortho* or *para* oriented to the amide. The reaction networks possible under these assumptions are given in Figure 8.3.

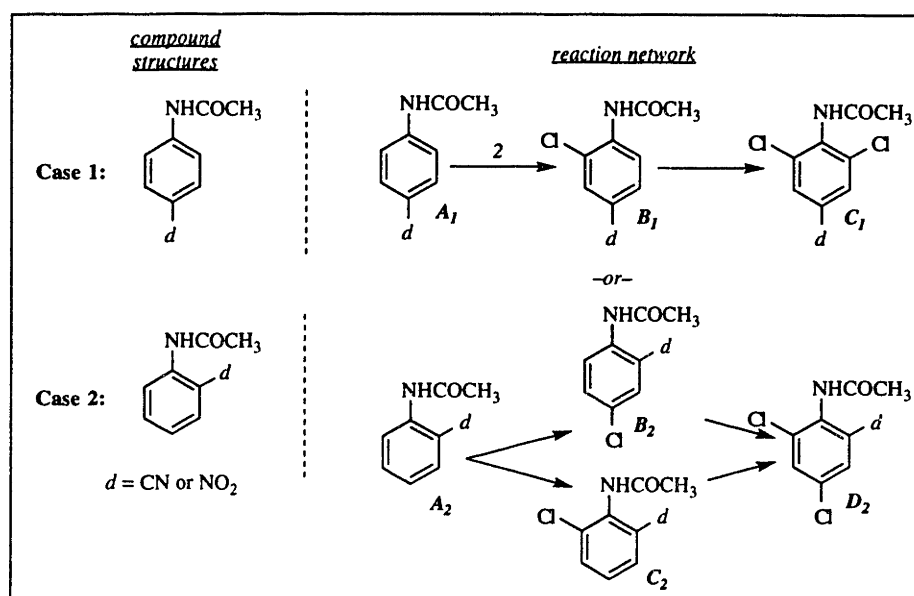
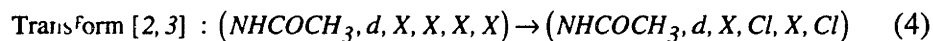
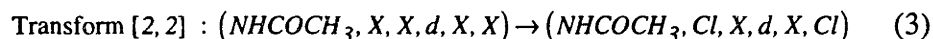


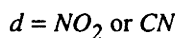
Figure 8.3. Reaction Networks Yielding Multiple Substitution that Should be Considered as Part of the Reaction Prediction Problem for Chlorination.

A_1 , A_2 , B_1 etc. are compound identifiers (used below).

The implied chemical transformations are given in Eqs. 3 and 4.



with

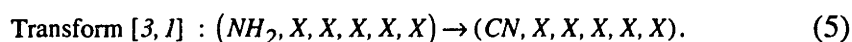


Note that the two assumptions are not reaction rules. If promising reactions are identified that seem to violate the assumptions, then one can reformulate the analysis and re-test such proposed reactions. In no way are such reactions being excluded *a priori*.

By an analysis similar to the one presented in chapter two, one concludes that to within the validity of the assumptions, very high yields of the monosubstituted compound can be achieved with minimal polychlorinated compound formation. Despite the low expected yield of polychlorinated compounds, the large number of chlorinated aromatics currently regulated by the EPA (Environmental Protection Agency) suggests that their formation at any concentration should be scrutinized. To insure that such compounds are identified, their formation is assumed to occur at a token level 1% per compound through inclusion of transforms [2, 2] and [2, 3]. Such a low level of formation is well within the "noise" of the predictive model.

8.3.3. Sandmeyer Reaction (Polysubstitution Transform Network)

The synthesis of benzonitriles by conversion of the corresponding diazonium salts is widely applicable to the class of nitro- and chloro-substituted compounds under consideration (Hagedorn, 1985). In every case, the diazonium salt should be formed directly from the amine and held in solution without intermediate drying. The net chemical transformation is simply



A number of competing reactions can potentially occur during diazotization or the Sandmeyer reaction, and undesirable reaction by-products can result if certain reaction rules are not obeyed. An example of one possible by-product class are the benzotriazoles, which are known to form in the diazotization of ortho phenyldiamines (Hertel, 1985; Figure 8.4).

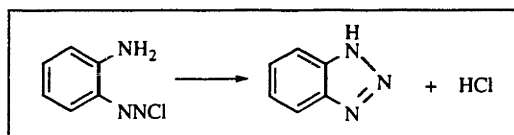
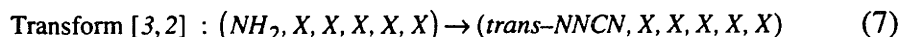


Figure 8.4. Benzotriazole Formation (Hertel, 1985).

Given the likelihood of complex reaction mixtures associated with phenyldiamines, a reaction rule is imposed precluding the formation of phenyldiamine precursors during retrosynthetic analysis. Following the notation in chapter two, given two functional groups, group_{*i*} and group_{*j*}, the logical form of the proposition is given in Eq. 6.

$$\text{Reaction Rule } [3, 1] : \left\{ \begin{array}{l} \{(\text{GROUP_TYPE}[\text{group}_i] \neq \text{amine}) \text{ OR} \\ (\text{GROUP_TYPE}[\text{group}_j] \neq \text{amine}) \} \end{array} \right\} = \text{True} \quad (6)$$

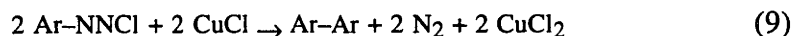
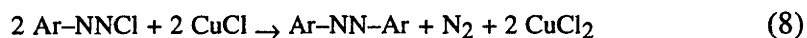
The Sandmeyer reaction also yields by-products. Isomerization of the diazocyanide adduct results in yield lost to the *trans*-diazocyanide (Eq. 7).



Based on literature accounts for related compounds, a yield loss of 20 [mole]% is assumed (Hegarty, 1978).

As discussed in chapter two, the diazotization and Sandmeyer reactions are not described well by linear free-energy relationships. For the compounds under

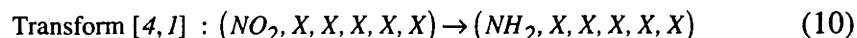
consideration, however, many results (Hagedorn, 1985) show that the reactions can be assumed to proceed rapidly and to consume essentially all of the reactants. The major issue with these compounds is the formation of by-products due to transforms such as Eq. 7. Additional by-products can result due to copper catalyzed diazo coupling reactions. Two commonly encountered side reactions are given in Eqs. 8 and 9.



Reactions such as Eqs. 8 and 9 are assumed to occur independently of substituent effects and are handled more efficiently in the process specification phase than in chemical conversion analysis.

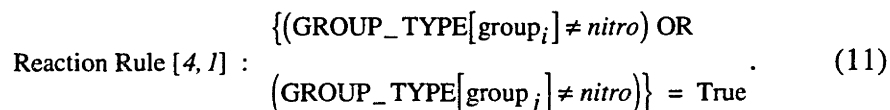
8.3.4. Béchamp Reduction (Monosubstitution Transform Network)

The Béchamp reduction is one of the available methods for transforming the nitro group to the amino group (Eq. 10).



One might be concerned about the reduction of the cyano group to aldehydes or other by-products on nitro-benzonitrile compounds. The literature indicates, however, that the Béchamp reduction tends to be highly selective to the nitro group in most cases (Hagedorn, 1985).

The reaction products can be difficult to predict if multiple nitro groups are present. In some cases both nitro compounds are reduced completely whereas in other cases one is selectively reduced (Shreve, 1963). In order to avoid this ambiguity, a reaction rule precluding formation of dinitro precursors is included:



An additional justification is that the formation of phenyldiamines as a chemical precursor is precluded by reaction rule [3, I]. Thus, Eq. 11 prevents the formation of many by-products that cannot be used to form final products.

Having excluded dinitro-substituted compounds, the main by-products formed are small quantities of amides due to reduction of the cyano group. Such by-products are most efficiently handled in the process specification phase rather than in chemical conversion

analysis. As in the case of the Sandmeyer reaction, the raw material is assumed to react completely.

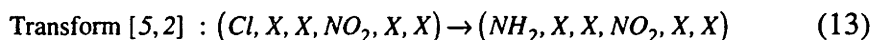
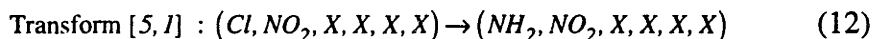
8.3.5. Additional Reactions

Several additional chemical reactions have been included for completeness without a detailed analysis required for reaction prediction. An ammonolysis reaction (an example of nucleophilic aromatic substitution; Baudet, 1924) of the form shown in Figure 8.5 has been included.

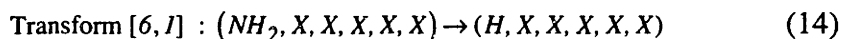


Figure 8.5. Ammonolysis Reactions.

It should be noted that partial rate factors are available for ammonolysis of activated substrates such as compounds with nitro and cyano substituents at the two and four positions with respect to chlorine (for example: Miller, 1968). The corresponding chemical transformations are given in Eqs. 12 and 13.



Hydro-dediazotiation, comprising the replacement of a diazonium ion by hydrogen (Hagedorn, 1985), has been included to consider reaction paths that involve removal of an amino group. The appropriate chemical transform is given in Eq. 14.



This type of transformation is challenging for several reasons. First, the corresponding retrotransform yields precursors that are more, not less, substituted. Hence, less substituted products have a larger number of potential reactions by the retrotransform. As a result, the size of the reaction path network that must be considered must grow in order to sufficiently simplify the intermediates to realistic raw materials. The resulting reaction paths are of interest because the transform allows directing groups to be introduced temporarily, thus providing a basis for highly integrated reaction paths and novel sources of intermediates.

8.4. Computational Strategy

Given a set of reaction prediction parameters (*e.g.*, relative rate bounds, pathway selectivity bounds, *etc.*), the algorithm developed in chapter two may be used for the synthesis of a reaction path network consisting of the specified target molecules (Figure 8.2) and chemical conversions (section 8.3). In practice one must anticipate several possible outcomes:

- a small number of promising routes is identified;
- a very large number of routes is generated, due to a missing reaction rule(s) or too relaxed an initial choice of reaction prediction parameters; and
- too small a number of routes is generated due to over-constraining reaction rules or reaction prediction parameters.

The uncertainty in the outcome reflects limits in the quality of information and degree of specification of the problem. One might argue that these limits discourage formal reaction path synthesis. To the contrary, in this work we adopt the position that the opportunity provided by reaction path synthesis to interactively determine these specifications in fact captures some of the essence of the reaction path screening problem in the first place. In particular, only a limited number of topological rules can be stated about any system superstructure *a priori*. Each reaction and each compound in a reaction path synthesis network leads to hypotheses that can be tested in the laboratory or through literature research to produce new reaction rules. Such rules augment the "general" reaction rules and may be highly decisive to the problem at hand (in principle, such an analysis can also provide a basis for model refinement). A further source of new rules may be practical considerations. For example, compound classes that are known to be more expensive than the products can usually be excluded as raw materials (but not necessarily as intermediates in a larger overall network).

In addition to development of new reaction rules, one can readily perform a variation on the reaction prediction parameters to determine which provide the strongest control over the number of results. A knowledge of which parameters are most significant can help to identify the most discerning measurements and experiments.

Based on the considerations presented in the previous two paragraphs, the following computational strategy has been adopted for the problem at hand. First, nominal values for the reaction prediction parameters are assigned and the initial computational results scrutinized for additional reaction rules. Second, a preliminary analysis of the chemical conversions is performed to identify promising reaction path configurations to effectively categorize the reaction path network in several smaller, more manageable pieces (no arrangement, however, is excluded from the final "production runs"). Finally, a variation in the values of the nominal parameters is performed to identify possible

modifications to the parameters. The computational strategy employed is summarized in Figure 8.6.

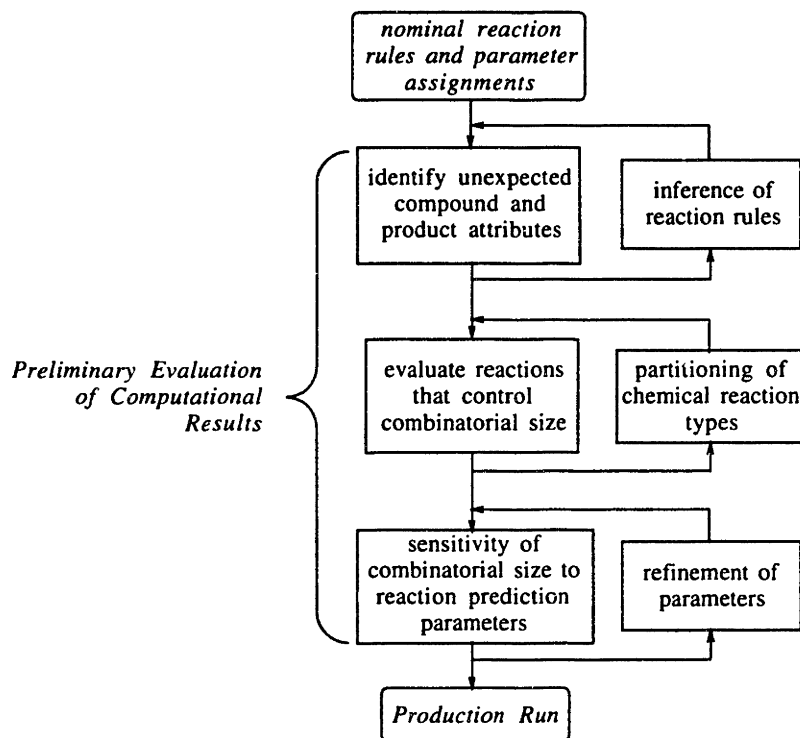


Figure 8.6. Computational Strategy Employed to Improve the Results Obtained from the Reaction Path Synthesis Algorithm.

8.4.1. Inference of New Reaction Rules

The set of chemical transforms under consideration is summarized in Table 8.3.

Table 8.3. The Set of Chemical Conversions and Their Underlying Chemical Transformations.

<i>Chemical Reaction</i>	<i>Retro-Transforms</i>	<i>Transforms</i>
nitration	[1, 1]	[1, 1]
chlorination	[2, 1]	[2, 1]–[2, 3]
Sandmeyer reaction	[3, 1]	[3, 1]–[3, 2]
Béchamp reduction	[4, 1]	[4, 1]
ammonolysis	[5, 1] and [5, 2]	[5, 1] and [5, 2]
hydro-dediazotation	[6, 1]	[6, 1]

It is also noted that the two reaction rules, [3, 1] and [4, 1], which limit the number of nitro and amino groups, are included.

Initial choices for nominal reaction prediction parameters are given in Table 8.4.

Table 8.4. Nominal Values for Reaction Prediction Parameters.

<i>Parameter</i>	<i>Value</i>	<i>Comments</i>
relative rate bound (R^{LO})	10^{-5}	set to lower bound on relative rate of nitration
product fraction bound (S^{LO})	0.01	heuristic based on cost of separation
pathway selectivity (π^{LO})	–	–
maximum no. steps (N^{MAX})	5	heuristic based on existing commercial processes

The relative rate bound is the one established for nitration under the assumption that the additivity treatment is subject to considerable deviations. Applying this bound to chlorination minimizes the chances of rejecting a feasible chemical conversion. The use of a single parameter for each of the reactions also facilitates the parametric analysis of this bound. The product fraction bound is set to a lower bound which will be varied later to determine its significance. Finally, the upper bound on the number of chemical

conversions, or "steps", in a reaction path helps to eliminate lengthy paths that are unlikely to lead to a practical chemical process.

Application of the retrosynthetic analysis algorithm yields 179 unique compounds (including the product molecules) for the set of five products. The results include several categories of reaction paths. First, a number of existing industrial routes are identified. One such route is the two-process reaction path shown in Figure 8.7 (Werner and Groggins, 1958). This route exploits ammonolysis of a dichloronitrobenzene in *Process A* to generate the chloronitroaniline intermediate. In practice, the chloronitroaniline serves as the feed to the next process (*Process B*), which uses the Sandmeyer reaction sequence and the Béchamp reduction to obtain the desired benzonitrile (Hagedorn, 1985).

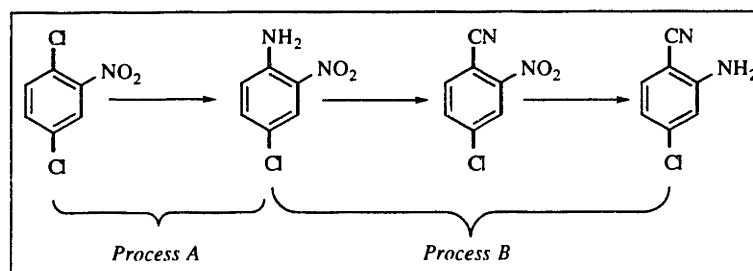


Figure 8.7. Existing Industrial Route Identified by Reaction Path Synthesis Algorithm. In practice the route is distributed over two processes (*Process A* and *Process B*).

Additional paths are also identified that serve to integrate several product lines. An opportunity for using chloroaniline precursor is given in Figure 8.8. The two paths may be suitable for implementation in a single plant and have the advantages of a potentially cheaper raw material source.

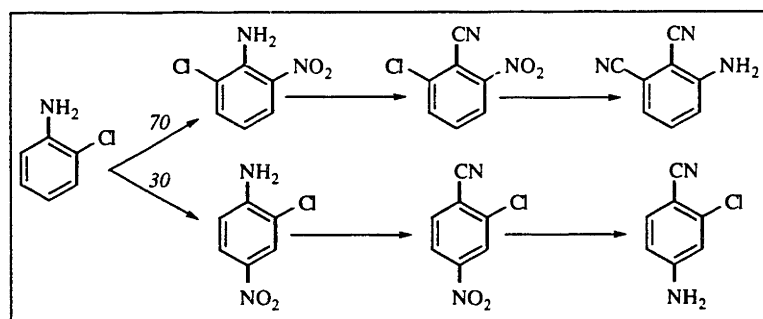


Figure 8.8. Alternative Route Involving a Chloroaniline Precursor.

Obscuring further analysis of the results are a large number of routes which originate with trichlorinated, tetra-substituted raw materials. A typical example of such a route is given in Figure 8.9.

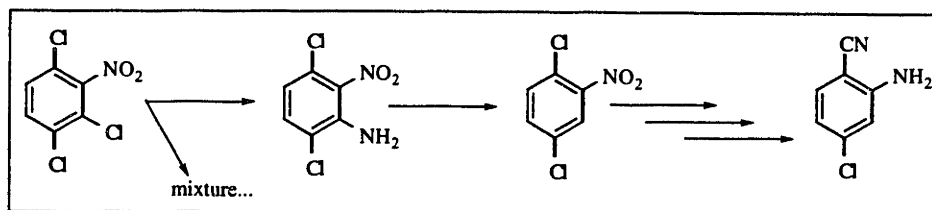


Figure 8.9. Candidate Route Involving a Polychlorinated Raw Material.

Several characteristics are observed:

- the trichlorinated precursors should be more expensive than any of the dichlorinated intermediates;
- from the products one can infer that trichlorinated precursors will involve at least two ammonolysis transforms, whereas existing routes for the products start with the appropriate dichlorinated precursor and use only one ammonolysis reaction; and
- the hydro-de-diazonation transform is used, which tends to increase the total number of chemical conversions required to obtain realistic raw materials.

There are several types of reaction rules useful for eliminating the reaction paths with these attributes. One approach would be to restrict the number of ammonolysis reactions in any one reaction path. Such an approach does not, however, eliminate all of the unrealistic raw materials. A more direct approach is to eliminate the class of unrealistic precursors by placing general structural constraints on the compounds that can be formed in the reaction path network. Based on the above considerations, two new reaction rules are inferred. First, trichlorinated compounds are eliminated from consideration (Eq. 15).

$$\text{Reaction Rule [5, 1]} : \left\{ \begin{array}{l} \{(\text{GROUP_TYPE}[\text{group}_i] \neq \text{chloro}) \text{ OR} \\ (\text{GROUP_TYPE}[\text{group}_j] \neq \text{chloro}) \text{ OR} \\ (\text{GROUP_TYPE}[\text{group}_k] \neq \text{chloro}) \} = \text{True} \end{array} \right. \quad (15)$$

Second, an additional rule is imposed to eliminate dichlorinated compounds with two or more nitro and amino groups (Eq. 16).

$$\text{Reaction Rule [5, 2]} : \left\{ \begin{array}{l} \{(\text{TOTAL_NITRO} = 0) \text{ OR} \\ (\text{TOTAL_AMINO} = 0) \text{ OR} \\ (\text{TOTAL_CHLORO} < 2)\} = \text{True} \end{array} \right. \quad (16)$$

Augmenting the original problem with Eqs. 15 and 16 reduces the size of the reaction path network (as measured by the number of compounds) from 179 to 125 compounds.

8.4.2. Partitioning of Chemical Reactions

A sizable fraction of the reaction path network is induced by the hydro-dediazotation transform. Elimination of the hydro-dediazotation transform reduces the size of the network, in fact, from 125 to 56 unique compounds. As mentioned above, the hydro-dediazotation retrotransform results in precursors of higher substitution than the product itself. Due to the larger number of retro-transforms needed to simplify these precursors to economical raw materials, the majority of routes involving hydro-dediazotation are not industrially competitive with routes that employ only simplifying retro-transforms. Despite this trend, however, it is not desirable to eliminate the hydro-dediazotation transform *a priori* because (i) it allows systematic modification of active sites on the substrate; and (ii) it provides an additional class of intermediates for the integration of reaction paths.

The application of the reaction path network algorithm is therefore partitioned into two stages. First, the reaction path network without considering the hydro-dediazotation transform will be created and characterized. Second, the reaction path network including hydro-dediazotation will be considered. While not all of the network is expected to be useful, it is of particular interest to gain insight as to what global attributes of such a network might justify the hydro-dediazotation transform.

8.4.3. Refinement of Reaction Prediction Parameters

Several variations have been performed to determine the strength of reaction prediction parameters and reaction rules. A summary of the results is given in Table 8.5.

Table 8.5. Results from Variation in the Reaction Prediction Parameters.

case description	max. no. steps	min. rate bound	prd. fraction bound	pathway select.	reaction rules	no. unique comps.
<i>no reaction rules</i>	5	$1.0E-5$	0.01	0	-	609
<i>no reaction prediction</i>	5	0	0	0	[3-4, 1]	238
<i>no rxn. pred. or rules</i>	5	0	0	0	-	692
<i>base case</i>	5	$1.0E-5$	0.01	0	[3-4, 1]	179
<i>new reaction rules</i>	5	$1.0E-5$	0.01	0	[3-4, 1], [5, 1-2]	125
<i>no hydro-dediazotation</i>	5	$1.0E-5$	0.01	0	[3-4, 1], [5, 1-2]	56
<i>increased no. steps</i>	7	$1.0E-5$	0.01	0	[3-4, 1], [5, 1-2]	165
<i>modified parameters</i>	5	$1.0E-5$	0.05	0.01	[3-4, 1], [5, 1-2]	125

The case in which no reaction rules are included shows the true size of the combinatorial network. The results indicate that the original reaction rules constraining the number of nitro and amino groups are essential. These rules combined with the additional reaction rules developed above are very effective in containing the combinatorial size of the problem, both for the nominal five steps and as the number of allowed steps is increased. Somewhat surprising is the strength of the original product fraction bound. In the final case, the product fraction and pathway selectivity bounds are increased to a more realistic

minimum level below which (i) neither the linear free-energy relationship nor the additivity principle is reliable; and (ii) the low yield of the product is unlikely to justify its separation and recovery. Turning to the other reaction prediction parameters, it is clear that the original product fraction and pathway selectivity bounds were too low. Increasing them to a more realistic value does not appreciably increase the number of unique compounds. A reduction in the pathway selectivity below the original value, however, considerably increases the number of alternatives.

Based on the above considerations, the production runs use the modified parameters incorporating the additional reaction rules, both with and without the hydro-dediazonation transform.

8.5. Production Results

8.5.1. Excluding Hydro-Dediazonation

The computations excluding hydro-dediazonation yield a relatively high fraction of reasonable chemical production routes. Individual chemical conversions involving most of the 56 compounds can be readily tested at the laboratory scale with a fair degree of confidence. The remaining chemical conversions are not practical for a variety of reasons, but are few enough in number that they can be discarded by inspection. The most promising reaction paths are listed in Figure 8.10.

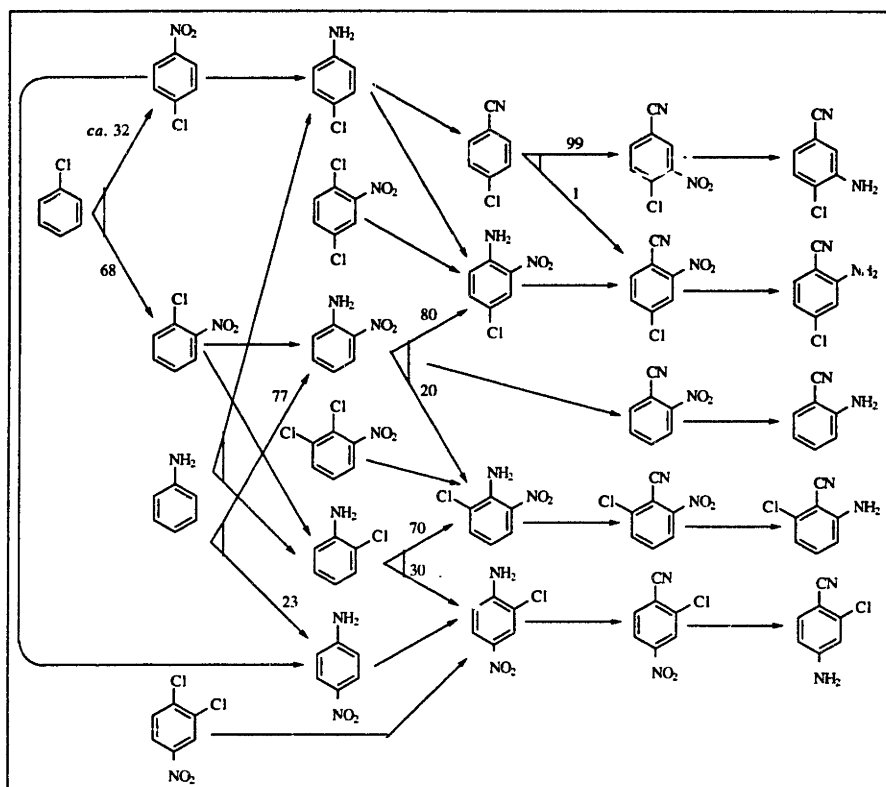


Figure 8.10. Benzonitrile Dye Intermediate Series Not Including the Hydro-Dediazotiation Transform.

Figure 8.10 contains quite a few routes that have been or are currently used in industry, including the class of routes based on ammonolysis-derived chloronitroanilines (depicted in Figure 8.7). Additional routes in this class are highlighted in Figure 8.11.

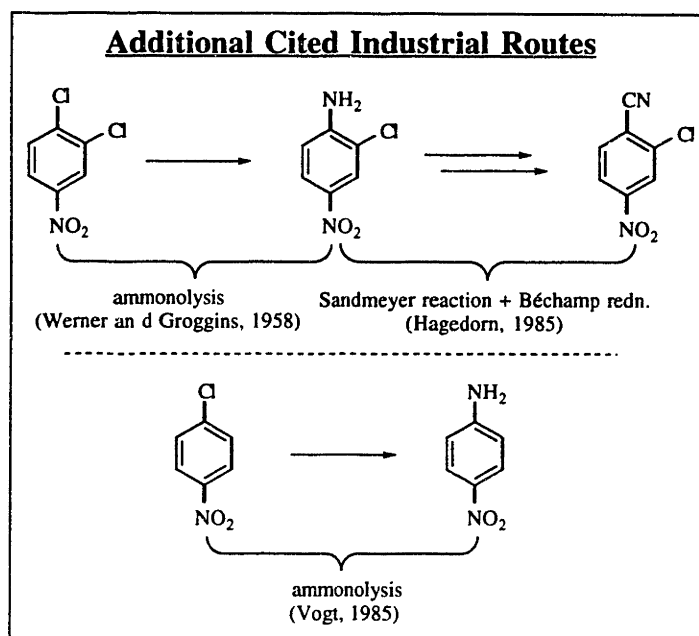


Figure 8.11. Existing Industrial Route Identified by Retrosynthetic Analysis Algorithm.

The pattern "(1) ammonolysis; (2) Sandmeyer reaction; (3) Béchamp reduction" used in Figure 8.11 should be transferable to the applicable syntheses in Figure 8.10. Not identified in Figure 8.10 are industrial routes to several precursors involving fairly well-known transformations that were omitted from the problem statement. One example is the ammoxidation transform, *i.e.*, direct conversion of a chlorotoluene to a chlorobenzonitrile (Figure 8.12).

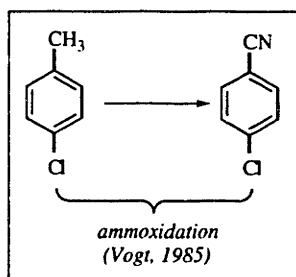


Figure 8.12. Industrial Ammoxidation Route to the 4-Chlorobenzonitrile Precursors in Figure 8.10.

Classical chemical conversions such as that shown in Figure 8.12 are readily identified by consulting industrial chemistry references (*e.g.*, Ullmann's Encyclopedia of Industrial Chemistry) or chemical databases.

So there is no lack of current strategies for the synthesis of these compounds. One of the potential drawbacks of the current industrial routes, however, is that their reaction paths are distributed over several processes. The producer of the benzonitriles lacks control over the cost of producing the raw materials and, in particular, must pay the premium on any new environmental costs on the chlorinated aromatic raw materials. In the case of ammonolysis, the processes for moderately activated substrates such as chloronitrobenzenes already have relatively high capital and operating costs—a typical ammonolysis reaction for dichloronitrobenzenes occurs in a titanium reactor at 4-5 MPa / 200°C and requires 10 hr (Vogt, 1985). On top of these traditional costs, the by-product precipitate salt (sodium chloride) must be disposed at a hazardous waste site, where costs can range from fifteen cents to the order of one dollar per pound.

The reaction path network (Figure 8.10) provides several alternatives that might give the manufacturer more control over these costs. While the economic scale of the ammonolysis process is often too large (and too far away from the final product) to be incorporated within process boundaries, several of the nitrations and chlorinations might be satisfactorily incorporated. A realistic set of alternative raw materials and reaction paths has been extracted from Figure 8.10 and depicted in Figure 8.13.

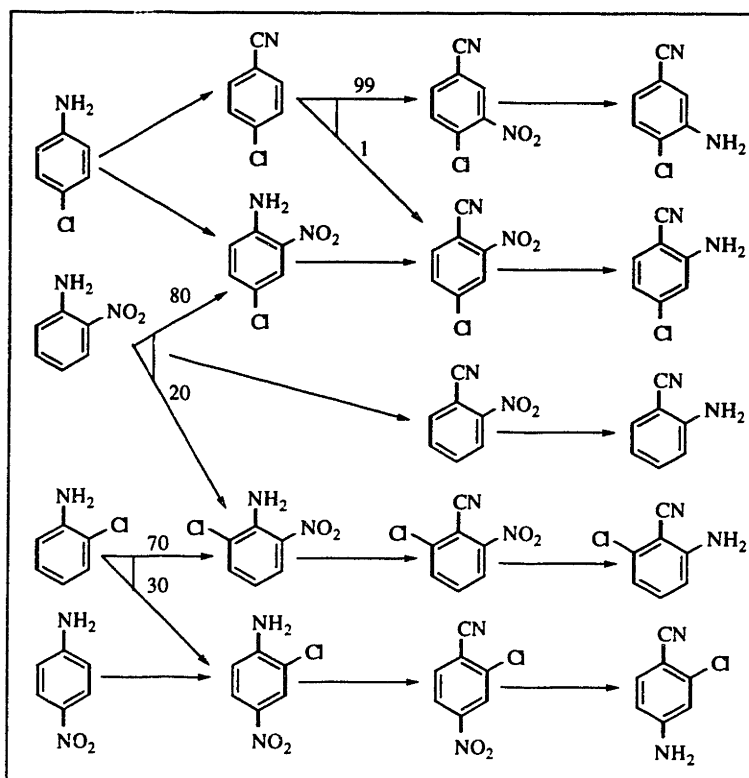


Figure 8.13. *New Reaction Paths of Potential Industrial Interest. These paths may allow the manufacture of key raw materials to be incorporated within the process boundaries.*

By exploiting the relatedness of the benzonitrile products, some of these paths may in effect allow two processes with two traditionally distinct set of waste streams to be folded into a single process. In these paths the need for high selectivity in generating chloronitroanilines is potentially reduced because common precursors such as *o*-nitroaniline are identified. Furthermore, the chloronitroanilines, which have been traditionally purchased as raw materials, can instead be produced within the process from 2-chloroaniline, 4-chloroaniline, 2-nitroaniline, or 4-nitroaniline.

A possible criticism of integrating the chemical reaction paths is lessened operating flexibility in the presence of periodic or uncertain product demand. Additionally, some flexibility in controlling the product variability may be lost. To some extent these limitations could be avoided through the integration of operations, scheduling, and inventory management in a multipurpose chemical facility. In this regard, a useful feature of the reaction path network is that it provides a basis for developing a superstructure of the available production configurations, which can be thought of as a primitive representation of available process recipes. For example, Figure 8.14 outlines the use of two alternate

configurations over several campaigns to meet peak demand for different products over time.

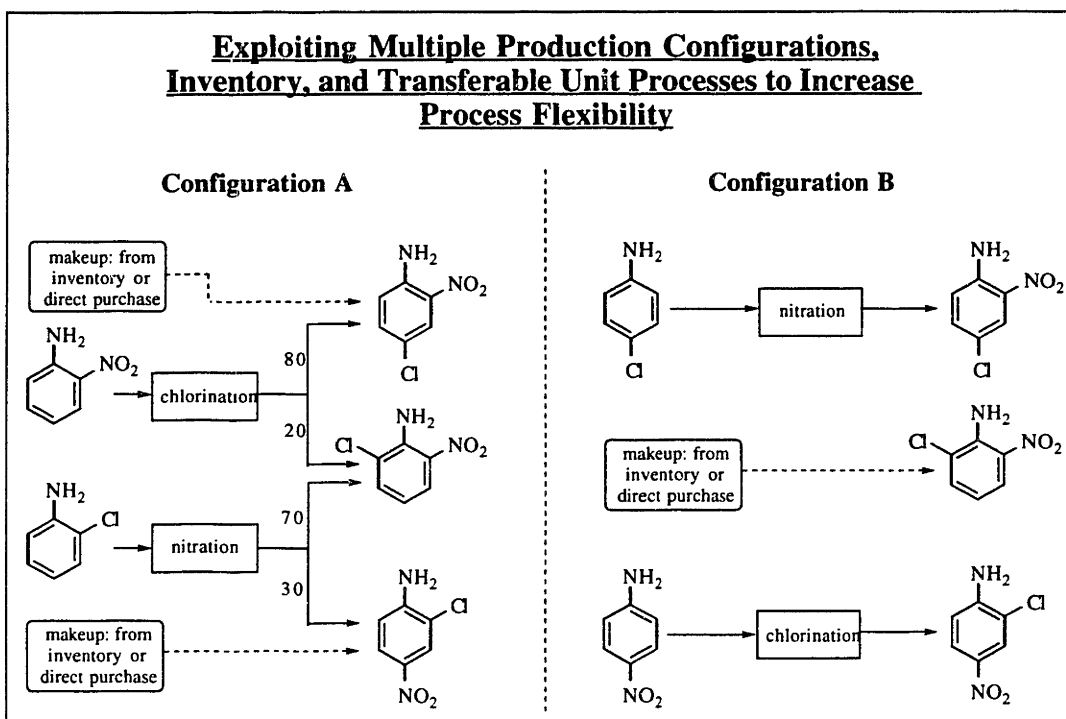


Figure 8.14. Superstructure of Alternate Paths that Generates Operating and Inventory Alternatives. Such alternatives are essential to achieving flexibility in the production schemes.

Both configurations are capable of exploiting very similar nitration and chlorination processes. Thus the two configurations could be implemented in a single existing multipurpose batch or semi-continuous facility as alternate campaigns. The first configuration is well-suited to meeting the peak demand of 2-chloro-5-nitroaniline (and the corresponding benzonitrile). The production of the other two compounds is fixed by the split fractions on the two reactions (in practice, there should also be some flexibility in the operating conditions to adjust the split fractions). If additional amounts of the intermediates are needed they can be drawn from inventory or purchased directly if it is logistically feasible to do so. Such an approach could adjust production for cyclical variations in product demand. Similarly, a strategy could be developed using configuration B for peak production of either 2-chloro-4-nitroaniline or 2-nitro-4-chloroaniline.

8.5.2. Incorporating Hydro-Dediazonation

As discussed previously, incorporating the hydro-dediazonation retrotransform significantly increases the size of the reaction path network (from one involving 56 to one involving 125 compounds). Three major factors account for the increased size:

- the hydro-dediazonation retrotransform yields a precursor molecule than the target molecule, hence introducing the need for additional retro-transforms to simplify the precursor;
- the hydro-dediazonation retrotransform introduces an amino group, which can effectively reverse the set of sites most susceptible to substitution in subsequent retro-transforms such as nitration, chlorination, *etc.*; and
- quantitative criteria are not employed to evaluate each proposed hydro-dediazonation, thus there is a tendency to generate all permutations of the reaction (subject to the reaction rules constraints, of course).

An interesting aspect of the reaction path network is that the simplest benzonitrile product, namely, 2-aminobenzonitrile, yields the largest number of proposed routes as it initially contains the largest number of hydrogen atoms. As shown above, however, the reaction rules do eliminate a large number of the permutations by limiting the number of nitro and amino groups.

Hydro-dediazonation chemistry potentially allows for even higher degrees of integration than achieved in Figure 8.10. Figure 8.15 illustrates an additional portion of the reaction path network in which the hydro-dediazonation transform plays a key role. Although certain aspects of the chemistry as well as the processing stability of the tetrasubstituted compounds are questionable, the reaction path network avoids the relatively expensive and low yielding Sandmeyer reaction, avoids a chlorinated raw material, and utilizes all major by-products. Thus, this example illustrates the potential combinatorial complexity that can be assimilated by the computational strategy.

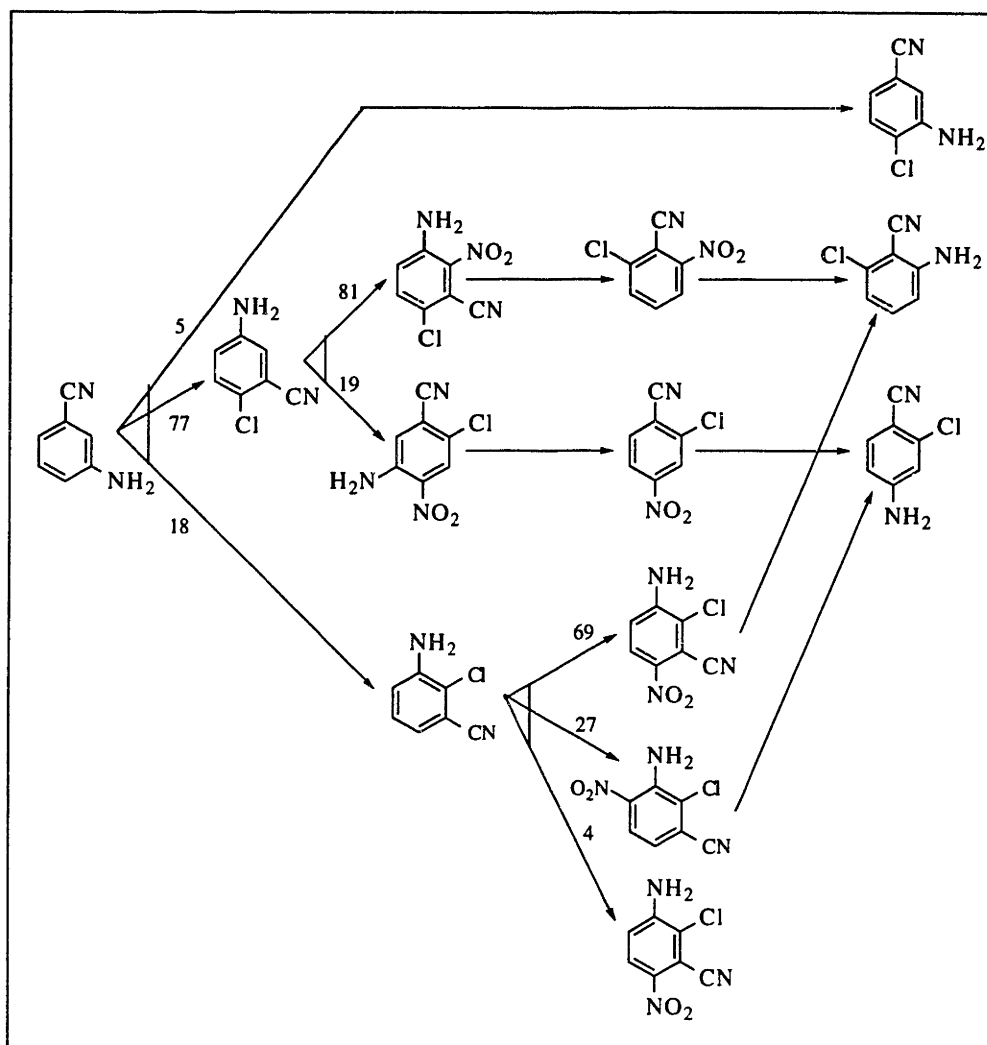


Figure 8.15. Highly Integrated Production Scheme that Exploits Hydro-Dediazotization Transforms

8.6. Future Experiments and Areas for Problem Refinement

A number of potential problems with the individual chemical conversions have been averted through careful selection of the chemical transformations, reagents (*e.g.*, choice of solvent, reagents), reaction rules, and quantitative screening criteria. If one is to consider enough alternative chemical conversions to identify novel reaction path networks, however, one must almost always make assumptions that can only be proven later in the laboratory. Thus, in addition to the introduction of new reaction rules, analysis of the

reaction path network serves to pose hypothesis that can be tested in the laboratory. A typical analysis might proceed along several general lines of inquiry, such as

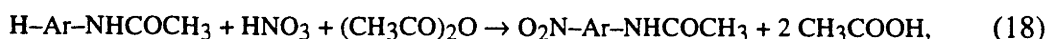
- identification of hydrolysis or other types of conversions between functional groups and solvents, solvent degradation products, or reagents to yield undesirable by-products;
- determining the effect of degradation products on solvent polarity and other factors that affect chemical reaction rates, selectivity, by-product formation, *etc.*; and
- assessing thermal stability of polysubstituted compounds containing reactive groups.

As potential problems are identified several possible lines of recourse are available. For classical chemical conversions such as nitration and chlorination, a number of variations have been developed, ranging from low temperature conversions (*e.g.*, to reduce the rate of hydrolysis of a compound) to completely different solvents and reagents. One should therefore consult the relevant chemical and patent literature for the class of compounds (or even the specific compound) under consideration. If a promising candidate is identified that affects the same basic conversion, then the basic integrity of the reaction path network can be preserved, though the rate, split fractions, and yield may have to be modified. Such changes are viewed not as a deviation from the reaction path synthesis strategy but as an integral part of the strategy. For very specialized chemical structures and conversions, it may turn out that the proposed chemical conversion is either manifestly infeasible or will require development of essentially new chemistry. If other components of the reaction path network are capable of replacing such a conversion, then it may be desirable to eliminate the problem conversion from the network.

Analysis of the two main reaction path networks (Figures 8.13 and 8.15) reveals several areas for further literature research and laboratory testing. First, tests are needed to determine how the by-product acetic acid affects the nitrations to be carried out in the acetic anhydride medium. In particular, the protective group chemistry for the amino group generates acetic acid as a by-product:



The product of (17) then reacts by a mechanism that involves both acetic anhydride and nitric acid:



to yield a total of three moles of acetic acid for every mole of raw material. One would like to minimize the need for intermediate capture or separation of the acetic acid until the final product is ready to be recovered as well. Measurements due to Paul (1958) show that the reaction rate is preserved for mixtures of up to 50 wt. % acetic acid (*ca.* 2 M HNO₃, 0.2 M

benzene, 25°C), although a significant reduction in rate is observed at higher concentrations; the reaction did not proceed to a detectable extent in pure acetic acid.

A second area of concern is the loss of the nitrile group due to reaction with acetic acid during several of the nitration conversions. Under acid or base hydrolysis conditions and with sufficient thermal agitation, such substitution is known to occur readily. In the case of 4-chlorobenzonitrile (Figure 8.13), protection of an amino group is not at issue. For this case, literature is available describing an alternate nitration that avoids the problem. The problem is more significant for the network depicted in Figure 8.15. Possible examples are shown in Figure 8.16.

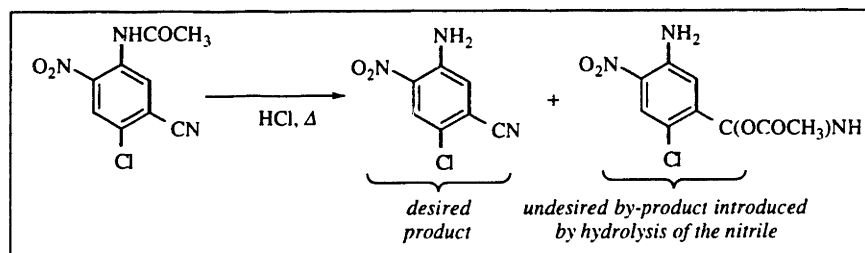


Figure 8.16. Example of the Potential For Yield Losses Due to Hydrolysis of the Nitrile.

The situation, however, is not clear-cut since the literature indicates that the presence of a *meta* amino group prevents hydrolysis of the cyano group (Dey and Doraiswami, 1933). It would be of interest to perform experiments to (i) evaluate the inhibiting effect of the amide group on the hydrolysis of the cyano group; and to (ii) test the inhibiting effect at the ortho and para positions. In addition, the presence of the cyano group might allow the nitration to be carried out under modified conditions without the use of protective chemistry.

8.7. Discussion

The case study has considered a fine chemicals manufacturing problem in which products are manufactured on a relatively small scale due to specialty of application as well as fluctuating and uncertain product demand over time. Adopting waste reduction and other environment-related strategies based on either (i) outsourcing the chloronitroaniline intermediates; or (ii) developing entirely new synthesis routes are both challenging problems. In particular, it is not easy for a specialty firm to assess the future environment-related premium it may have to pay for the outsourced intermediates. Additionally, specialty firms tend to rely on niches of expertise and process technology. Balancing the potential reduction in environment-related costs, a company must consider the cost associated with minimizing product variability, transferability of the process technology, and a whole host of other issues.

Synthesis and analysis of the reaction path network has uncovered a third alternative that could help to address these challenges. Rather than continue to outsource the intermediates, it may in fact be desirable to shift production of chloronitroaniline intermediates *into* the process by using less selective, readily available nitration and chlorination reactions of nitro- and chloroanilines that yield several useful by-products. Any firm involved in the production of nitriles or other dye-related compounds should have considerable knowledge of these chemical conversions. With such knowledge, the approach allows the existing process to be augmented with several additional unit processes.

An important set of issues that must be addressed are the processing implications associated with integrated chemical reaction paths. Whereas traditional process integration (*e.g.*, heat exchange networks and separations systems) tends to complicate the scheduling problem and reduce overall process flexibility, integration of the chemical reaction paths can reduce certain aspects of the process complexity. For example, the production of several intermediates from a single chemical conversion effectively merges several reaction and separations tasks. As one might expect, the simplification of the process recipe tends to occur at the expense of new coupling within the process technology. Process technology tradeoffs to be addressed in preliminary process screening include the higher cost of separations associated with this strategy.

The coupling introduced by the reaction path networks motivates a head-to-head comparison with existing industrial routes. At some point one might expect the set of separations and interdependencies between reaction paths to become so complex as to make actual implementation prohibitive. A related example is the development of heat exchanger networks that are so integrated that the required piping scheme would be unable to meet safe piping standards. In general, for either of the two reaction path networks (Figures 13 and 15) one may anticipate that computer-aided operations are essential to successfully implement the intricate inventory and campaign strategies associated with such networks, particularly if the product demand varies over time (Figure 8.14).

A limiting case for the generation of highly integrated reaction path networks is encountered when one introduces retro-transforms that increase the level of substitution of the precursor over that of the target molecule. For such transforms, the network complexity is inversely proportional to the product complexity. Many of the proposed reaction paths are manifestly infeasible due to the chemical properties, cost, or number of chemical conversions associated with the proposed intermediates. Certain paths, on the other hand, involve novel intermediates that could be very difficult to identify without a systematic reaction path strategy (*e.g.*, the tetrasubstituted intermediates in Figure 8.15). In this work we have attempted to reduce the number of false-positive results associated with these transforms through introduction of reaction rules eliminating manifestly infeasible intermediates (*e.g.*, requiring too many steps to convert to inexpensive raw

materials). In general, a basic direction for further research involves development of reaction rules that can be applied to the reaction path network as a whole. The underlying difficulty with incorporating such rules arises from the complexity and symmetry of the molecular structures. Development of logical rules through introduction of binary variables, for example, tends to yield dense mathematical constraints whose number is proportional to the number of compounds in the network.

8.8. Conclusions

Reaction path networks have been developed for simultaneous manufacture of five related benzonitriles. In the first network, existing industrial routes based on ammonolysis of dichloronitrobenzenes are identified as are several new synthetic strategies. In particular, a strategy for incorporating the manufacture of outsourced intermediates within existing process boundaries *via* nitration and chlorination of chloroanilines and nitroanilines has been developed. The strategy could potentially allow a reduction in raw material costs as well as in certain environment related costs. In a second network, the role of the hydrodediazonation transform in controlling network complexity is explored. A key feature of the transform is that it yields precursors of higher substitution with substitution sites of reversed activation. As a result, choices of chemical intermediates yielding highly integrated reaction paths have been identified that would be difficult to find without a systematic reaction path synthesis strategy.

The first reaction path network (Figure 8.13) appears to be the most promising for rapid implementation. Several steps towards process development should be taken. First, the choice of solvents and operating conditions need to be tested in cases where the nitrile group is present. Second, preliminary process screening in the form of chemical property estimation and estimated material balances is needed to facilitate unit process network synthesis as the next step towards process development.

8.9. Supplement: Calculation of Partial Rate Factors

The partial rate factors have been calculated from a combination of Hammett-type parameters, experimental measurements for ortho:para and meta:para ratios, and direct system measurements. This section provides the raw data tables and references required to perform the computations.

8.9.1. Nitration

Table 8.6. Raw Data and Calculation of Partial Rate Factors for Nitration (HNO_3 , Ac_2O , 25°C). T refers to Taylor (1990). S refers to Stock (1968)

Subst.	References	fitted params.			exptl. params.		partial rate factors		
		ρ^+	σ_m^+	σ_p^+	$\frac{1}{2}o:p$	$\frac{1}{2}m:p$	r_o	r_m	r_p
							<i>calculated</i>		
NHAc	T:299,458; S:68	-6.0	0.08	-0.62	1.7	—	8921	0.33	5248
CN	T:463,486	-6.0	—	0.66	4.3	20.2	9.28E-5	4.36E-4	2.15E-5
NO_2	T:463,486	-6.0	0.73	0.79	10.7	—	2.88E-6	4.17E-5	2.69E-7
							<i>measured</i>		
Cl	T:463	-6.0	—	—	—	—	0.03	8.4E-4	0.13

8.9.2. Chlorination

Table 8.7. Raw Data and Calculation of Partial Rate Factors for Chlorination (Cl_2 , AcOH , 25°C). T refers to Taylor (1990). S refers to Stock (1968).

Subst.	References	fitted params.			exptl. params.		partial rate factors		
		ρ^+	σ_m^+	σ_p^+	$\frac{1}{2}o:p$	$\frac{1}{2}m:p$	r_o	r_m	r_p
							<i>calculated</i>		
Cl	T:463	-10.3	—	0.115	0.29	0.01	0.019	6.54E-4	0.0654
CN	T:395,463	-10.3	—	0.66	4.0	12.7	5.38E-7	1.71E-6	1.35E-7
NO_2	T:395,463	-10.4	—	0.79	5.9	27	6.28E-9	3.03E-8	1.12E-9
							<i>measured</i>		
NHAc	S:43,46	-10.3	0.08	—	—	—	6.1E5	0.15	2.5E6

Chapter 9

Integration of the Modeling Strategy with Process Scaleup Considerations: Separation of Reaction By-Products.

9.1. Introduction

Separation and recovery of reaction by-products are key challenges in industrial research and development. With higher waste disposal costs, strategies for increased recovery of by-products are potentially attractive for the minimization of waste streams. Current emissions and waste recovery regulations, however, mandate expensive recycling steps or reductions in many of the solvents and mass separating agents used in recovery processes. Furthermore, the set of design tradeoffs to be addressed are not confined to the separation system but can also influence the overall process flowsheet, particularly the design and operation of reaction and recycle systems.

Computer-aided design of by-product recovery systems is desirable early in process development to establish whether separation can be accomplished at an economic cost. Process simulators allow both the preliminary design of certain separations (*e.g.*, distillation columns) and the simulation of a process flowsheet taking into account overall equipment requirements and material and energy balances. Additionally, equation-based modeling methods such as dynamic simulation allow simulation of many batch and semi-continuous unit operations. Application of these tools early in process development allows a more systematic consideration of alternate designs and their implications, both in the individual separations and at the flowsheet level. Furthermore, the computational results can be refined through selective experiments developed from previous insights.

One of the key challenges in any process modeling effort is the determination of chemical properties required for design. As a general rule, by-product separation systems are particularly challenging due to the diversity of organic compounds that can be

encountered. Whereas the thermophysical properties of common solvents and mass separating agents are well-modeled by pure component group contributions such as Joback's methods (Reid *et al.*, 1987) and activity coefficient models such as UNIFAC (Fredenslund *et al.*, 1976), their accuracy varies considerably among specialized compound classes, particularly those exhibiting intramolecular bonding effects and containing multiple combinations of functional groups. A contributing factor to these limitations is that group contribution methods tend to average out variations in functional groups on different molecules, resulting in potential underestimation of differences in properties between stoichiometrically similar compounds. A classic example of such effects are isomers of polysubstituted aromatic compounds. While most group contribution methods do not distinguish certain isomeric forms, in many instances the isomers can exhibit significantly different pure component properties. Additionally, correcting for these differences within group contribution methods can introduce a considerable number of new parameters due to the large number of possible isomers.

In chapters four through six we have proposed a chemical modeling strategy that exploits molecular calculations as a basis for better incorporation of structural information in pure component and certain mixture property models. In particular, the method computes electronic density and molecular surface area measures to account for variations in functional groups that are traditionally averaged out by group contribution methods. Such measures are directly applicable to estimating differences in properties between aromatic isomers. In addition, the method possesses features generally desirable for model building in industry's research and development, including

- a reduced number of geometric parameters as compared to group contribution methods;
- capability for refinement based on new measurements or more detailed molecular calculations; and
- transferability of optimized model parameters to the estimation of additional related properties.

As an illustration of the method, we present a case study for the separation of aromatic isomer mixtures derived from the results in chapter eight. In particular, two key separations of chloronitroaniline isomers have been identified that could lead to significant process improvement *via* in-sourcing the production of these intermediates. The general production strategy for the synthesis and separation of the two mixtures is depicted in Figure 9.1.

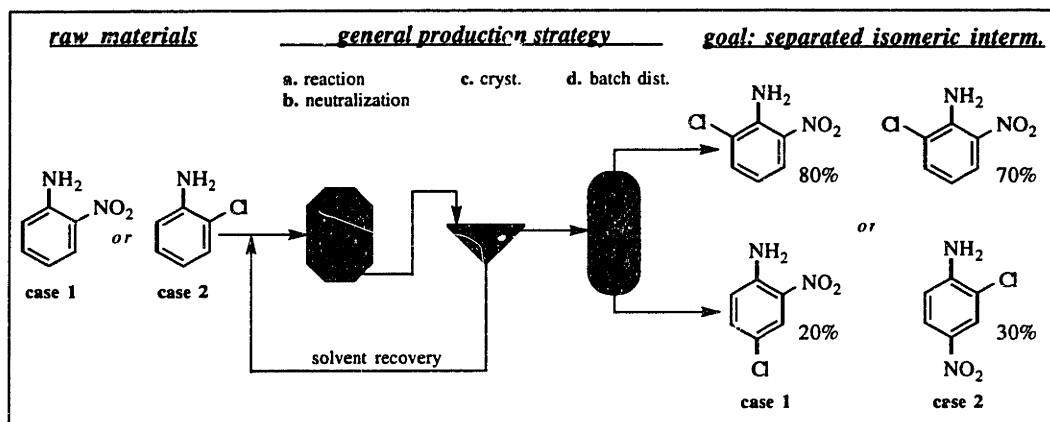


Figure 9.1. Model By-Product Recovery Problems Involving Mixtures of Aromatic Isomers.

The first three tasks, a–c, are documented in several patents for compounds in the chloronitroaromatics class (e.g., Smythe *et al.*, 1991). The last step, on the other hand, requires vacuum distillation of a mixture of two isomers. There is little evidence on the feasibility of this separation, mainly because the two intermediate compounds are traditionally out-sourced and produced separately by ammonolysis of dichloronitrobenzenes (Baudet, 1924). Essential in the evaluation of distillation alternatives is an estimate of the separation factor as a function of system temperature. However, current physical property models using group contribution methods are unable to distinguish between these compounds since they are composed of the same set of functional groups.

Key questions to be addressed by application of the prototype simulation module include the following.

- Are the differences in thermophysical properties of the aromatic mixtures large enough to allow separation by vacuum distillation?
- Can the distillation be accomplished well below the flashpoint and well above the melting points of the compounds?
- Can the thermodynamic property estimates be obtained in a computationally efficient manner to allow a large number of compounds to be treated?

This chapter addresses these issues for the estimation of separations factors in the model example depicted in Figure 9.1. First, a brief overview of the computational chemical modeling strategy is presented to illustrate its application within a property prediction model. This model is then incorporated within a prototype property estimation module. The module is used to estimate the normal boiling points for a set of compounds, including those shown in Figure 9.1. Based on the computational results, the property simulation is utilized to determine separation factors as a function of temperature for the compounds of interest. These results are then incorporated into a batch distillation column model.

9.2. Overview

A prototype property estimation module incorporating geometric and structural information *via* molecular calculations is presented. The module is illustrated in a case where traditional group contribution methods significantly underestimate differences between structurally similar compounds. The proposed property module utilizes efficient computational chemistry tools developed in chapters five and six to derive estimates of the normal boiling point and critical volume. These results are then used to obtain pure component vapor pressures by a combination of group contribution and corresponding state methods. The approach is illustrated to determine the relative volatilities of mixtures of chloronitroaniline isomers. Within the benzonitrile case study (chapter eight), these separations are in fact critical to the ability to in-source the production of the isomeric intermediates. It is shown that the isomeric mixtures should have good separation factors *ca.* 2.0 at temperatures well below the flashpoints of the compounds. Predicted differences in the boiling points (*ca.* 15K) are significantly larger than the mean absolute error of the method (*ca.* 9K) and reproduce expected intramolecular effects resulting in boiling point depression. Based on the efficiency of the underlying computations, this approach may be particularly useful to assist in preliminary process screening before complete thermophysical measurements are available.

9.3. Property Estimation Module Prototype: Use of the Normal Boiling Point

9.3.1. Review of Computational Chemistry Tools

A class of property models based on a linear relationship between the property of interest and the intermolecular potential (Eq. 1) have been developed in chapters five and six.

$$F^m = \sum_{n=1}^N \alpha_n f_n[m], \quad (1)$$

where F^m is the property of interest for a compound, m , and the $f_n[\bullet]$ ($n = 1, \dots, N$) are functionals that define interaction energy contributions (*e.g.*, dispersion, polar, and hydrogen bonding) with weighting coefficients, α_n . The α_n are property-specific parameters that reflect the specific properties of interest as well as the thermodynamic state of the system (in particular the combinatorial entropy). The $f_n[\bullet]$, in contrast, are intrinsic descriptors of the intermolecular potential that can, in principle, be transferred to related properties *via* Eq. 1. The class of properties that can be treated by such relationships include those that are amenable to linear free-energy assumption. In other words, those

properties in which the combinatorial entropy contribution can be estimated through measurements for a reference compound. Examples include the normal boiling points, critical temperatures, and critical volumes of many organic liquids, as well as certain mixture and reactivity properties such as linear solvation-energy relationships (derivations given in chapter six).

Our approach to the construction of Eq. 1 has been to adopt a lattice potential representation of the intermolecular potential energy contributions. The degree of coordination between molecules has been described in terms of the total molecular surface area, $A(m)$, and the atomic contributions to the molecular surface area $A_c(m)$, $c \in C_m$, $C_m \subseteq C$ (set of all possible atom-types). For the athermal case, which is a first approximation, a suitable relationship for the degree of coordination between two atoms centered on different molecules is given by Eq. 2.

$$\theta_{c_1 c_2}(m) = p_{c_1}(m) p_{c_2}(m), \quad c_1, c_2 \in C_m \quad (2)$$

where

$$p_c(m) = A_c(m) / A(m), \quad c \in C_m, \text{ and} \quad (3)$$

$$A(m) = \sum_{c \in C_m} A_c(m). \quad (4)$$

Equation 2 is essentially the joint probability of two components in two non-interacting compounds being coordinated to one another. Based on Eq. 2, the lattice potential has been formulated as

$$U_{c_1 c_2} = z A(m) \left\{ \psi_{c_1} p_{c_1}(m) \right\} \cdot \left\{ \psi_{c_2} p_{c_2}(m) \right\}, \quad (5)$$

where z is the coordination number of the lattice and the product $\psi_{c_1} \psi_{c_2}$ is the ideal binary interaction between two isolated components. Hence, in this model the intermolecular potential contributions are represented by single component and perhaps binary parameters (binary parameters are employed only for a few specific interactions). The individual contributions to the intermolecular potential, $f_n[\bullet]$ ($n = 1, \dots, N$), are defined by breaking Eq. 5 into dispersion, polar, and hydrogen bonding contributions.

In the definition of the $f_n[\bullet]$, there are several opportunities for incorporating molecular calculations. First, optimization of the geometry by molecular orbital calculations provides a reference molecular structure that is reasonable for relatively small organic molecules (*e.g.*, mol. wt. < 200). From a knowledge of the molecular geometry

and individual atomic radii, the molecular surface area contributions can be computed directly. This approach contrasts with the approach employed in UNIFAC, wherein different functional groups are assigned fixed surface areas. The proposed approach is more flexible as a means of obtaining accurate surface areas and requires fewer adjustable parameters. A second application of molecular calculations is the incorporation of electron density measures into the interaction potential parameters. Such measures provide a way of accounting for structural effects such as resonance and intramolecular charge transfer interactions. These interaction can lead to significant variations in properties between molecules consisting of a similar set of functional groups.

Using the molecular calculation strategies described above, three functions corresponding to the dispersion, polar repulsive, and polar attractive contributions, are presented. The first function is a nonpolar dispersion interaction (Eq. 6).

$$f_1(m) = A(m) \quad (6)$$

The second function is a polar repulsive interaction energy contribution (Eq. 7),

$$f_2(m) = \sum_{c_1} \sum_{c_2} \left(b_2^{c_1} q_{c_1}(m) p_{c_1}(m) \right) \cdot \left(b_2^{c_2} q_{c_2}(m) p_{c_2}(m) \right), \quad (7)$$

$$c_1, c_2 \in C_m \text{ st. } \left(b_2^{c_1} q_{c_1}(m) p_{c_1}(m) \right) \cdot \left(b_2^{c_2} q_{c_2}(m) p_{c_2}(m) \right) > 0$$

and the third function is a polar attractive interaction energy contribution (Eq. 8).

$$f_3(m) = \sum_{c_1} \sum_{c_2} \left(b_2^{c_1} q_{c_1}(m) p_{c_1}(m) \right) \cdot \left(b_2^{c_2} q_{c_2}(m) p_{c_2}(m) \right), \quad (8)$$

$$c_1, c_2 \in C_m \text{ st. } \left(b_2^{c_1} q_{c_1}(m) p_{c_1}(m) \right) \cdot \left(b_2^{c_2} q_{c_2}(m) p_{c_2}(m) \right) < 0$$

where $q_c(m)$ is the partial atomic charge for a component $c \in C_m$ and b_c is the charge scaling factor.

Equations 6–8 have been used as a basis for development of a general model for prediction of the normal boiling point. Based on the correlation of approximately 135 diverse compounds, including a number of polysubstituted aromatics, the model shown in Eq. 9 has been developed.

$$NBP = 1.73[0.04]f_1(m) - 47.2[0.1]f_2(m) - 15.8[0.6]f_3(m) + 101[6], \quad (9)$$

where the numbers in square brackets are standard errors for the estimated parameters (chapter six). A hydrogen bonding parameter has also been estimated but is not included here because it is not used.

Given a model of the form of Eq. 9, a set of computational modeling tools has been developed for automated property prediction. The integrated computational elements of the method are given in Figure 9.2.

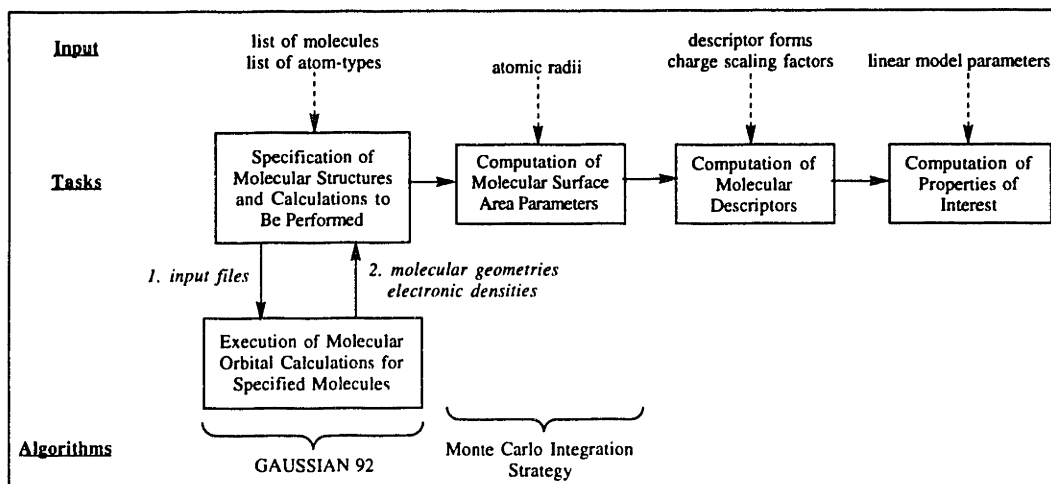


Figure 9.2. Computational Element for Estimation of the Normal Boiling Point.

Based on an initial specification of the molecules under consideration, a code has been developed to generate initial geometric configurations from AM1 optimized bond-lengths. A set of input files to the GAUSSIAN 92 molecular orbital code are submitted for execution. In this chapter, AM1 molecular geometry optimizations are utilized as a basis for the molecular geometry. Mulliken population analysis of the computed electronic densities are used to obtain the partial atomic charges. Upon completion of the calculations, these parameters are returned and submitted to the next stage of the computations. The next stage consists of computing the molecular surface area contributions for the set of molecules. Efficient numerical computation is based on a Monte Carlo strategy employing Hammersley-Wozniakowski sample points. These results are then combined with the partial charges and charge scaling factors according to Eqs. 6–8 to estimate the normal boiling point (Eq. 9). When the results of the molecular orbital calculations are available in advance, the execution times for the process is well under 1 CPU second per molecule on a DEC 3000 alpha workstation. When molecular orbital calculations are generated as an integral part of property estimation, the entire process run approximately 30–60 seconds per molecule on a Cray C90, with the bulk of the time obviously being taken up by molecular calculations. A detailed description of the computational framework and algorithms employed is given in chapter five. Values for the atomic radii and charge scaling factors are given in chapter six.

9.3.2. Components of a Property Estimation Module

Methods for the estimation of the pure component properties (*i.e.*, boiling point, critical properties, vapor pressures, *etc.*) are currently available in property estimation packages such as those employed in ASPEN and other process simulators. In general, the underlying models are interdependent on one another. For example, Joback's method (Reid *et al.*, 1987) for the estimation of the critical temperature requires the normal boiling point as input. As a second example, the estimation of pure component vapor pressures by the Gomez-Thodos method (Reid *et al.*, 1987) uses the critical temperature, critical pressure, and the normal boiling point as input. Consequently, the accuracy of property predictions are critically dependent on the initial properties supplied to the models.

This section proposes a structure for a property estimation module incorporating molecular calculations. The goal of this strategy is to obtain improved estimates of pure component properties in general by improving the accuracy of the normal boiling point. In particular, the molecular calculations are used to estimate the normal boiling point and the critical volume (computed using the equation derived in chapter five). An improved estimate of the normal boiling point is incorporated into Joback's method to estimate the critical temperature. The critical volume and the critical temperature are then used to estimate the critical pressure according to an equation given in Grigoras (1990). Finally, the normal boiling point, critical temperature, and critical pressure are incorporated in the Gomez-Thodos method to obtain an estimate of the critical temperature at one or more temperatures of interest. The components of the resulting property prediction module are depicted in Figure 9.3.

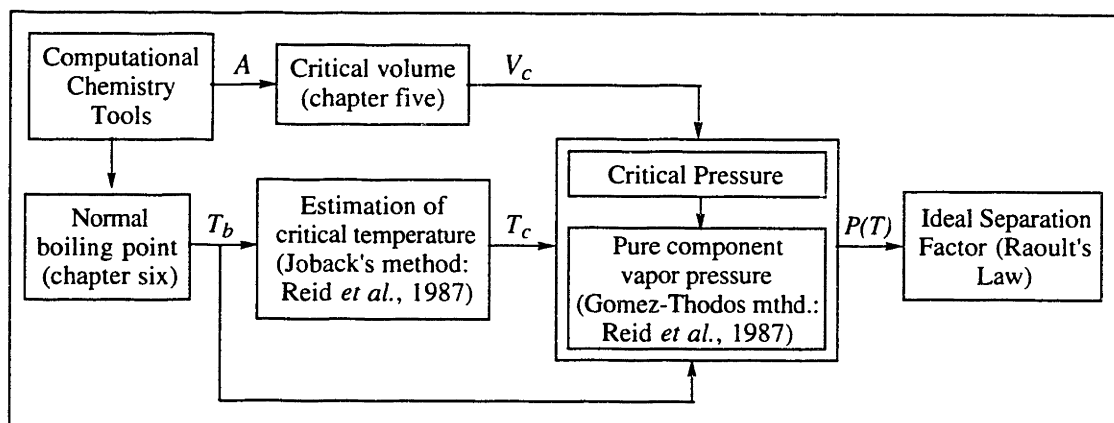


Figure 9.3. Prototype Assembly of New and Existing Computational Tools for Implementation in a Process Simulator Module. "A" is the molecular surface area; " T_b " is the normal boiling point; " T_c " the critical temperature; and " $P(T)$ " the pressure as a function of temperature.

In the case of aromatic isomers, the approach allows a knowledge of the difference in normal boiling points to be used to determine unique vapor pressures and therefore compute unique ideal vapor pressures.

9.4. Estimation of Relative Volatilities for Separation of Aromatic By-Products

The components of the property estimation module can be applied to preliminary analysis of the isomer separation problem outlined in the introduction. Normal boiling points have been estimated for a set of compounds in the reaction path network (chapter eight), include those depicted in Figure 9.1. Computational results are given in Figure 9.4 (intermolecular potential descriptors for these compounds are given in the compound databank).

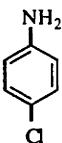
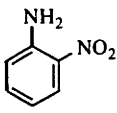
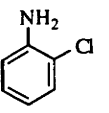
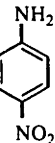

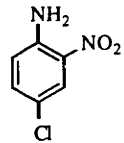
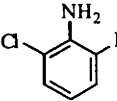
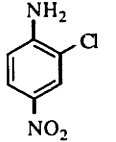
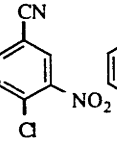
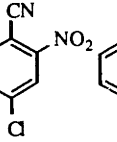
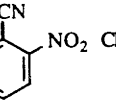
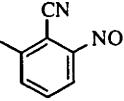
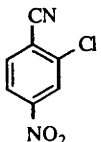
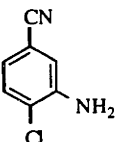
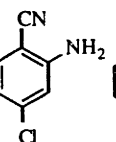
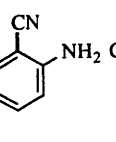
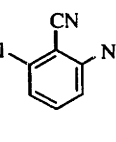
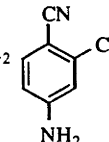
						
NBP (K):	484.8	553.3	472.5	572.4	497.6	562.0
NBP (Joback's mthd.):	478.3	591.0	478.3	591.0	529.2	635.1
T _c (Joback's mthd.):	726.2	796.5	707.8	824.0	724.8	796.0
						
	549.8	567.6	579.3	566.2	550.8	558.5
	635.1	635.1	687.6	687.6	643.4	687.6
	778.7	804.0	802.1	783.9	773.9	773.4
						
	573.5	553.2	544.9	518.8	533.5	544.5
	687.6	608.2	608.2	564.1	608.2	608.2
	794.1	781.8	770.2	745.1	754.1	769.6

Figure 9.4. Estimation of Normal Boiling Point (NBP) and Critical Temperature (T_c) for Compounds in the Reaction Path Network (Figure 8.13). The normal boiling point is estimated by the linear energy-property

model. Calculation of the NBP by Joback's method is included for comparison.

The main source of systematic deviations between the computed normal boiling points and those computed by Joback's method are large errors for aromatic nitro compounds by Joback's method. The predicted values for the normal boiling point and the critical temperature may be unrealizable for certain of the compounds due to decomposition.

From a knowledge of the normal boiling points and the critical temperature, estimation of the pure component vapor pressures and ideal separation factors (as a function of overall system pressure) can be computed. Such information is very useful in determining whether it is worthwhile to pursue separation of the compounds *via* vacuum distillation. The two key separations in the reaction path network involve the chloronitroaniline product mixtures. Figure 9.5 depicts the net desired result of a separations train for each of the separations.

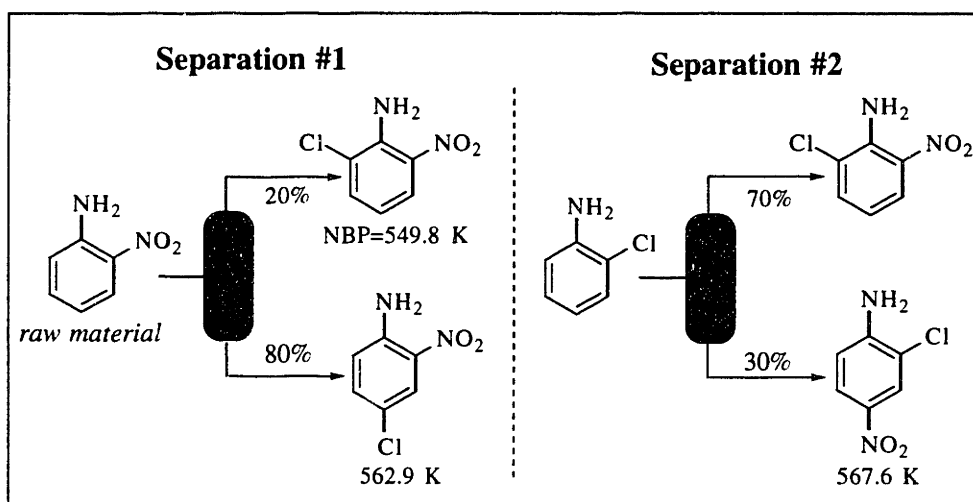


Figure 9.5. Main Separations of Isomeric By-Products To Be Accomplished.

In practice, accomplishing the separations to achieve a desired purity requires multiple distillations, recrystallizations, *etc.* Determining the optimal configurations of separation trains, operating pressures, and control profiles is a process synthesis problem that depends upon the degrees of purity required (*i.e.*, whether or not both products be recovered in high purity for further utilization). A pre-requisite to the use of process simulation or optimization tools is a minimal set of thermophysical data required to determine the separability factor as a function of temperature. In the design of a batch distillation column operated at a constant pressure (*e.g.*, under a vacuum), the separation factors must be computed at different compositions as the distillation progresses.

As the above calculations show (Figure 9.4), standard group contribution methods such as Joback's method (Reid *et al.*, 1987) do not distinguish between the isomers. Consequently, there is no basis to estimate separation factors using most of these methods. For the particular compounds of interest, there is some hope of separability by distillation due to the boiling point depression caused by intramolecular interactions between ortho-oriented nitro and amino groups. The calculations by the linear property model bear the effect out, predicting boiling point depressions of *ca.* 12 and 17 K in comparison to the other two forms of the isomers.

To determine the ideal separability of the two mixtures, the strategy depicted in Figure 9.4 has been implemented for the estimation of the pressures and relative volatilities as a function of temperature. Plots of the results are given in Figures 9.6 and 9.7. Melting points and flashpoints have been obtained from Booth (1985).

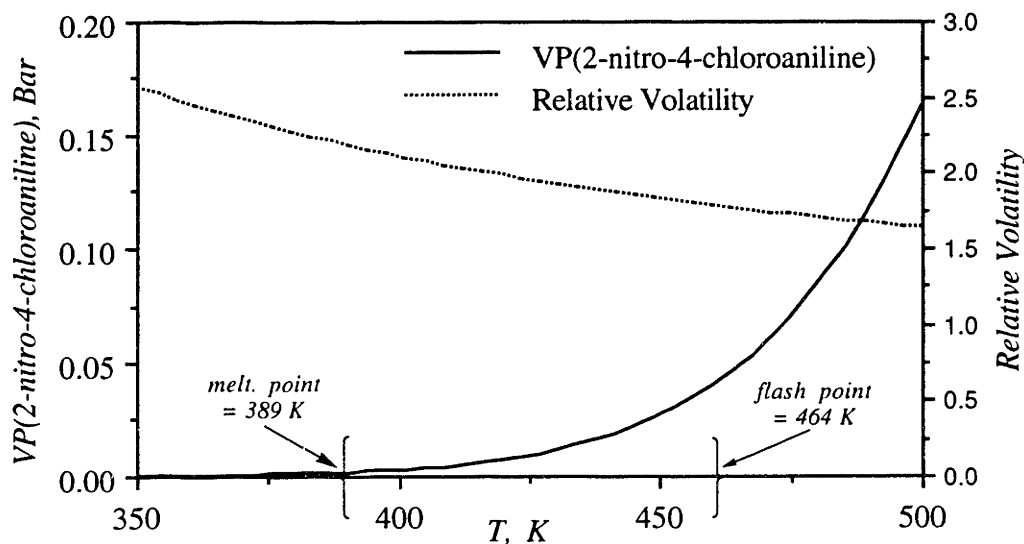


Figure 9.6. Plot of the Pure Component Vapor Pressure and Separation Factor in the First Separation Depicted in Figure 9.5.

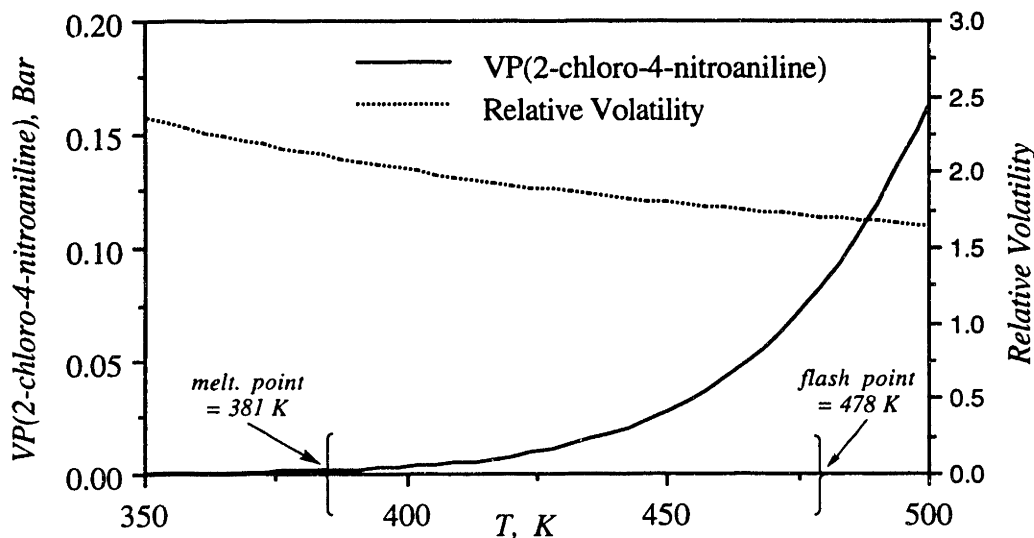


Figure 9.7. Plot of the Pure Component Vapor Pressure and Separation Factor in the Second Separation Depicted in Figure 9.5.

For the 2-chloro-6-nitroaniline / 2-nitro-4-chloroaniline mixture, the relative volatility falls in the interval [1.76(melting point), 2.18(flashpoint)] with a mean relative volatility of $\bar{\alpha} = 1.95$. For the 2-chloro-6-nitroaniline / 2-chloro-4-nitroaniline mixture, the relative volatility falls in the interval [1.71(melting point), 2.10(flashpoint)] with a mean relative volatility of $\bar{\alpha} = 1.88$. In both cases, the mean separation factors are in a range allowing use of batch distillation. Thus, the results indicate that separation by low pressure or vacuum batch distillation is indeed possible.

9.5. Application to Column Design

The estimated mean relative volatility for the 2-chloro-6-nitroaniline / 2-chloro-4-nitroaniline mixture has been used to estimate the design of a batch distillation column for the recovery of 2-chloro-6-nitroaniline as a major product. Batch columns are usually run under one of two policies. In the first policy, the overhead composition of the desired product is maintained constant by continually increasing the reflux ratio as the distillation proceeds. In the second policy, a constant reflux ratio is maintained and the overhead drawn until the average composition of the desired product drops from an initially high value to a target value. Often the distillate target is met before the desired bottoms composition is achieved. This event can occur because the batch distillation column does not contain a stripping section. As a result, obtaining the desired level of purity in the bottoms requires making one or more "slop cuts"—*i.e.*, the distillate is discarded into a second overhead receiver. In order to achieve good yields of the bottom product(s), slop cuts are often recycled into the still and mixed with the new batch charge. Such a recycling

strategy would be essential, for example, to recover the 2-nitro-4-chloroaniline from the 20% 2-chloro-6-nitroaniline / 80% 2-nitro-4-chloroaniline mixture.

The design calculations presented in this chapter adopt the second operating strategy, employing the algorithm developed by Diwekar and Madhavan (1991). A listing of the code is given in appendix five. Due to the time varying nature of the still pot composition, study-level design algorithms for a batch distillation column are somewhat more complex than for continuous distillation. The approach of Diwekar and Madhavan (1991) treats the design problem as a continuous rectification (*i.e.*, distillation with no stripping) with a time-varying feed. A block representation of the inputs and outputs is given in Figure 9.8.

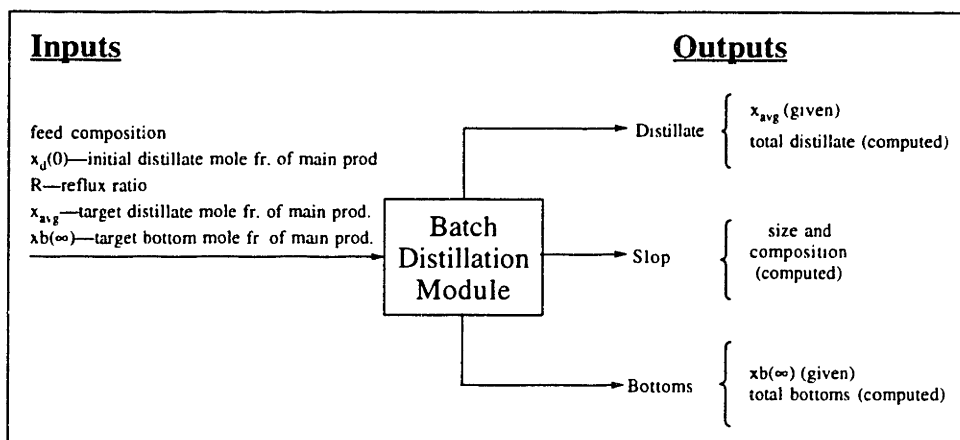


Figure 9.8. Block Representation of the Net Inputs and Outputs of the Computational Model for Batch Distillation.

The initial distillate mole fraction is an adjustable parameter that is used in determining the number of trays and, by implication, the amount of total distillate (this parameter must always be set larger than the target distillate mole fraction). Increasing the initial distillate mole fraction results in higher recovery of the main product.

Column design parameters for the 70% 2-chloro-6-nitroaniline / 30% 2-chloro-4-nitroaniline separation have been computed for three reflux ratios at an initial distillate mole fraction (of 2-chloro-6-nitroaniline) of 0.999, a target distillate mole fraction of 0.97, and a bottoms mole fraction of 0.05. Results are given in Table 9.1.

Table 9.1. Results for the Binary Distillation of a 70% 2-chloro-6-nitroaniline / 30% 2-chloro-4-nitroaniline Mixture. Algorithm due to Diwekar and Madhavan (1991) employed. An initial distillate mole fraction (of 2-chloro-6-nitroaniline) of 0.999 is employed.

case	reflux ratio	no. trays	total distillate(*)		total bottoms		total slop cut
			97% 2-chloro-6-nitroaniline		5% 2-chloro-6-nitroaniline		
1	2.0	22	0.54 (77%)		0.15		0.31
2	4.0	14	0.67(96%)		0.24		0.09
3	6.0	12	0.69(99%)		0.28		0.03

*Total recovery of the product

Plots of the 2-chloro-6-nitroaniline composition in the still and in the distillate product receiver are depicted in Figure 9.9.

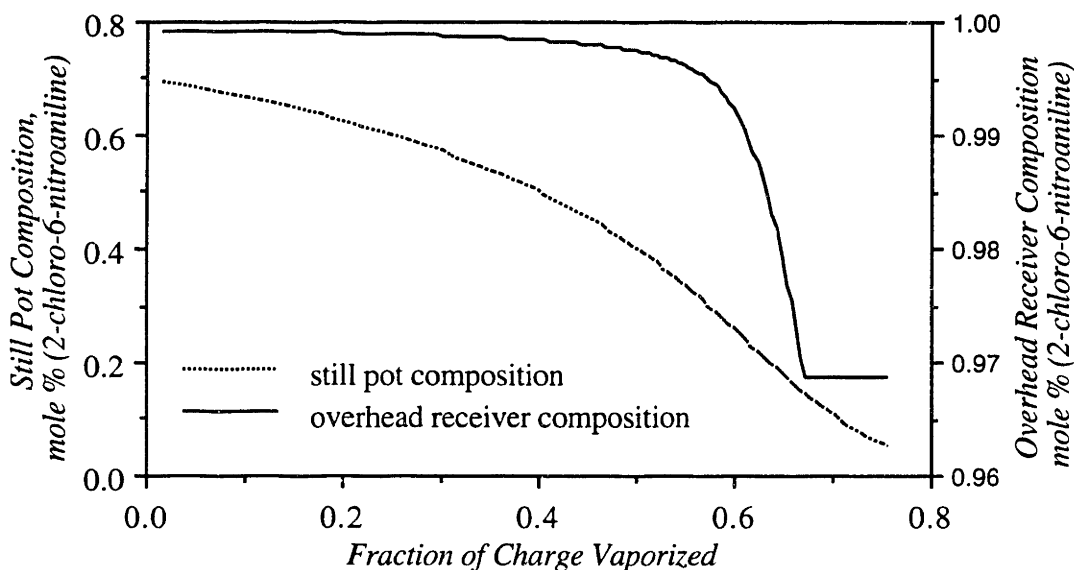


Figure 9.9. Theoretical Curves for Batch Distillation of a Binary 70% 2-chloro-6-nitroaniline / 30% 2-chloro-4-nitroaniline Mixture. The number of trays is 14 and the reflux ratio is 4.0.

9.6. Conclusions

A computationally efficient strategy employing molecular calculations has been demonstrated for pure component property estimation. It has been shown that the method is computationally fast, allowing property estimation of compounds for molecular weights of at least 50, typically within one minute. This result has led to elucidation and testing of key elements in a future property estimation module. Incorporation of the molecular calculations in a property prediction module allows compounds to be treated that are beyond the scope of traditional group contribution methods. In particular, estimation of

relative volatilities by the approach indicates that two mixtures of chloronitroaniline isomer reaction products are likely to be separable. The thermophysical results have been employed in a short-cut batch design module to estimate design parameters for one of the separations. These results validate the underlying strategy for the synthesis of property models developed in chapters five and six. Of particular importance is the improved ability to deal with separations of diverse reaction product mixtures.

Chapter 10

Importance of Considering Multiple Production Routes in Preliminary Process Screening

10.1. Introduction

A basic goal of process synthesis is the systematic comparison of design alternatives to find an improved or optimal process configuration. Achieving the goal often requires the integration of process technology from multiple chemical production routes. Any chemical industry is composed of strategic sets of chemistries and process technologies, both of which must be continually updated and improved to meet a wide range of market and regulatory changes. Such changes include newly discovered chemistry, evolving customer preferences, new patents, and new environment-related regulations, to name a few. Thus, selection of a new chemical production route should be viewed as an integral part of a larger strategic plan involving the entire set of production routes. Such a plan should focus on answering the following questions:

- does the new production route make appropriate use of existing technology, raw materials, and intermediates; and
- what does the new route bring in terms of transferable technology, more generally applicable process chemistry, and safety- and environment-related improvements?

In short, incorporation of a new production routes provides opportunities both to target desirable types of chemistry and process technology and to phase-out chemistries which are increasingly less desirable (*e.g.*, those which create high levels of waste, require inefficient unit operations, or which create special safety hazards).

Global integration and optimization of chemical production routes can lead to processes that are considerably different than isolated processes. For example, usable reaction by-products from one process may be merged with several production routes

(chapter nine). Resulting integrated processes might in fact lead to a lower overall level of waste generation than from unintegrated "cleaner" production routes. As a second example, process chemistry that exploits a raw material common to multiple production lines has been used as a basis to in-source production of several useful intermediates, thereby merging the manufacture of two production lines (chapter eight). In this context it should be noted that a series of related products is often manufactured within a single multipurpose plant. Multipurpose plants have several inherent advantages related to process scheduling, including abilities to exploit common process technology among several production routes and to adapt to cyclical demand patterns for products.

Whereas tools are available for the analysis and selection of individual unit operations, this thesis is one of the first to propose tools to compare alternate process chemistries taking into account the strategic role of multiple production routes. Chapters two and eight have demonstrated an algorithm that uses a knowledge of multiple products to identify a common set of raw materials, intermediates, and chemical conversions. Chapter nine implements novel modeling tools that evaluate key separations for judging the potential of several production routes. The present chapter presents a framework combining the two sets of results to examine the importance of multiple production routes in preliminary process screening. The general strategy taken is to optimize the net present value of a superstructure of production routes for given production constraints. Evidence for the need to consider multiple production routes is sought at two levels:

- (i) analysis of the degree of integration of resulting process configurations; and
- (ii) differentiation between limiting cases in which existing industrial processes and novel integrated processes are obtained from the superstructure.

The basis for the framework is the total plant synthesis strategy introduced in chapter seven. The strategy recasts the reaction path network as a superstructure of alternatives incorporating process-level features, including the representation of salable intermediates, waste streams, and equipment and utilities requirements. Mathematical programming techniques based on mixed-integer linear programming are employed to identify the optimal process configuration to within the limits of the mathematical models and design assumptions. Process components are incorporated through a unit process representation, wherein process requirements are centered around individual chemical conversions. Key topological and material flow design variables are treated explicitly in the formulation while remaining components of the unit processes are treated as scalable base case design estimates.

Such an approach is useful in addressing preliminary screening problems for two reasons. First, the quality of information is frequently limited to initial design estimates in the early stages of process development. To assist in deciding which new experiments are the most important, one would like to focus on the design issues that are decisive to the feasibility of a reaction path. Second, the unit process representation is a general one and

can be expanded to incorporate design estimates at many levels, ranging from short-cut designs to detailed process simulation results.

The benzonitrile study considered in chapters eight and nine provides a good basis for preliminary process screening. The study reveals a number of possible approaches for the integrated production of the five benzonitriles (chapter eight) and provides estimates of separation factors, which support the feasibility of separating two key aromatic isomer mixtures (chapter nine). A large number of process alternatives can be generated from the reaction path network (Figure 10.1) if one considers

- the production of the chloronitroaniline intermediates as intermediate products;
- hybrid in-sourcing and out-sourcing of different chloronitroaniline intermediates;
- the recovery and utilization of reaction by-products; and
- the optimization of production levels.

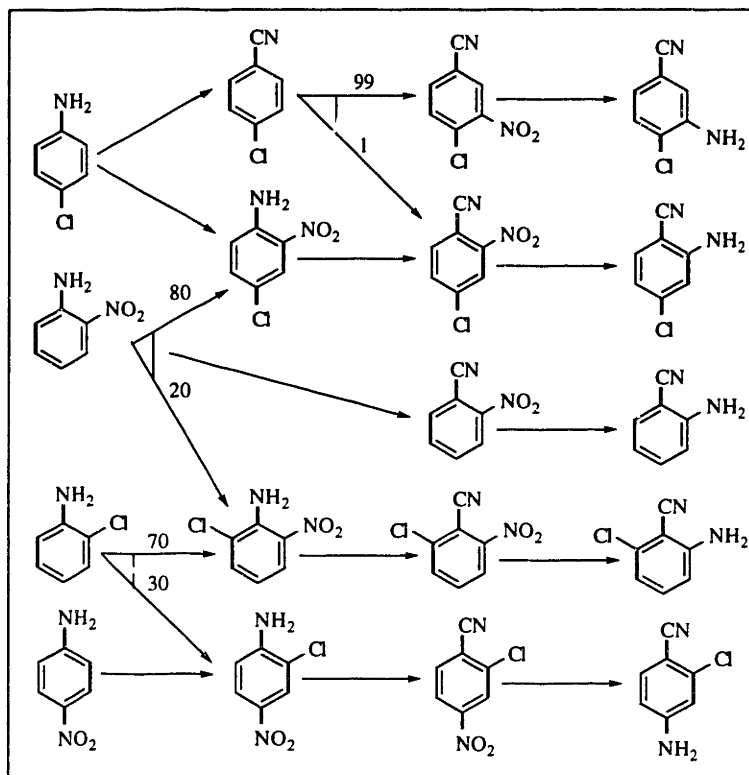


Figure 10.1. New Reaction Paths of Potential Industrial Interest. These paths may allow the manufacture of intermediates to be incorporated within the process boundaries.

The general strategy developed in chapter seven is exploited identify the most promising configurations. The effects of variations in nominal commodity prices and production costs are tested explicitly. Such variations serve a two-fold purpose. First, market conditions will in fact fluctuate over time. A key question to be addressed is as follows: are proposed configurations flexible and profitable in the presence of the fluctuations? In other words: can the initial process design adapt its production routes and equipment uses to these changes? Second, there is uncertainty in the underlying unit process designs due to their preliminary nature. Identification of key design components provides a good basis for experimental design and model refinement. For the deterministic analysis, four main calculations are performed:

- base case design derived from estimated production costs, supplies, and demands;
- variation in prices for intermediate products;
- modification of the products line to exclude on of the products; and
- a significant increase in waste disposal cost.

A second thrust of the case studies is to look at parametric uncertainty in production targets. Such uncertainty is expected to have a large impact on the integration of unit processes to reflect that certain unit processes are useful in several production routes. The chance-constrained programming formulation developed in chapter seven has been adapted to solve several relevant case studies.

As a final goal, the case study provides a good test of the mixed-integer linear mathematical programming formulation (chapter seven) since it yields a realistic-sized problem with a large number of binary variables. In chapter seven, strategies have been proposed to tighten the formulation of the mathematical program. Establishing the computational feasibility of the formulation is a pre-condition for integration with other process modeling and optimization tools (*e.g.*, process simulators). Key metrics used in the evaluation include the CPU times required to obtain optimal solutions, the difficulty in solving the relaxed linear programs, and the initial linear programming relaxation gaps.

The organization of the chapter is as follows. First, the unit process synthesis problem is posed and the salient elements of chapter seven reviewed. Second, production targets and design constraints are presented. Third, specific computational details are introduced and the computational results reported. Based on the computational results, insights specific to the problem and general implications are discussed. Finally, a sketch of how more detailed computational calculations can be incorporated are discussed. In a supplementary section, a listing of unit process specifications for the reaction path network is given.

10.2. Overview

The total plant synthesis strategy (chapter seven) is employed for preliminary process screening of the benzonitrile reaction path network (chapter eight). Several case studies are considered and results reported, including variations in waste stream costs, variations in product demand, and uncertainty in future production targets. It is shown that the reaction path network contains a number of potential strategies that might be subject to further analysis, with topological features including *(i)* recovery of in-sourced chloronitroaniline by-products for use in additional production routes; *(ii)* use of excess capacity to produce several intermediate products for eventual sale; and *(iii)* reduction in waste generation through minimization of parallel tasks. A main goal in the case studies is to demonstrate computational feasibility of the mathematical programming approach. The mathematical programming formulations on average consist of approximately 680 equations, 400 continuous variables, and 90 discrete variables (an example is given in appendix four). In each case, initial linear programming relaxations are within a few percent of the global optimum, with the optimum being achieved in several CPU seconds on a VAX mainframe computer. The result establishes a necessary criterion for the unit process synthesis technique being integrated with other process modeling tools.

10.3. Problem Overview

10.3.1. Notation

The unit process network is an interconnected set of unit processes, material and energy flows, and material storage nodes (one for each stream-type). Based on the assumption of unlimited storage, network material balances are written around each storage node to define net material and energy flows. The materials entering and leaving each unit process are defined in terms of internal material and energy balances, which can yield several streams depending on the internal separations configuration within the unit process. A summary of the material streams and their notation is given in Figure 10.2.

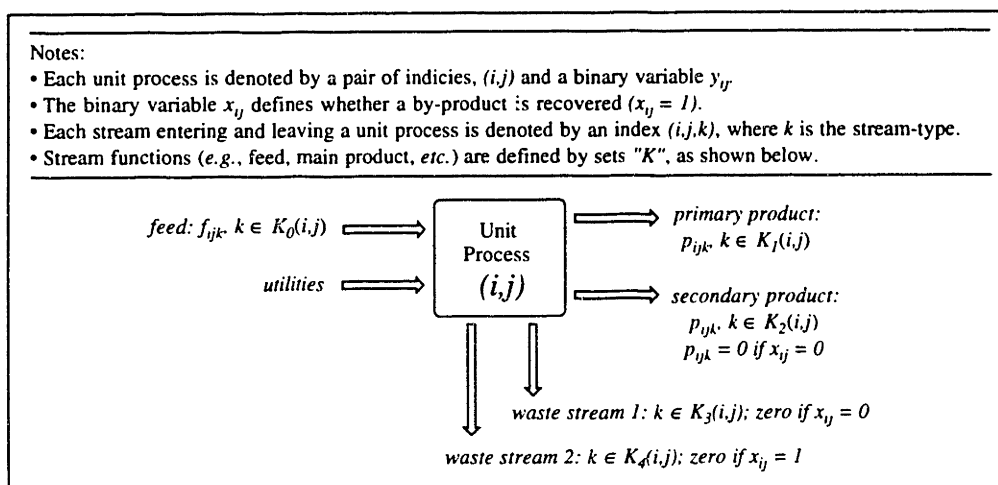


Figure 10.2. General Set of Material and Energy Flows Entering and Leaving Unit Process Boundaries.

Each unit process, denoted (i, j) , is assumed to accept a single type of feed f_{ijk} , $k \in K_0(i, j)$. Two types of material streams are produced by each unit process: product streams and waste streams. Product streams fall into two categories. The main product stream, p_{ijk} , $k \in K_1(i, j)$, is produced regardless of design decisions within the unit process. The secondary product stream, p_{ijk} , $k \in K_2(i, j)$, is produced only if additional recovery is performed. The choice of additional recovery is modeled by a binary decision variable, x_{ij} (Figure 10.2). Similar to the product streams, there are two types of waste streams, $k \in K_3(i, j)$ and $k \in K_4(i, j)$, which depend on the status of by-product recovery.

10.3.2. Network Specifications

Chapter seven has introduced several possible forms for a deterministic objective function. A general feature of the objective functions is the need to consider material, energy, and equipment requirements simultaneously. The evaluation function used in the case study is a net present value metric, which is to be maximized through the choice of unit processes, stream-types, and mass flows. Using the standard technique of Lang factors (Peters *et al.*, 1980) to account for the total capital investment and total product costs, the evaluation function is given by

$$NPV = 0.7I_T - 7.6C_E - (C_{RM} + C_W + C_U + 1.3C_L), \quad (1)$$

where NPV is the net present value measure, I_T is the total income, C_E is the cost of equipment, C_{RM} is the cost of raw materials, C_W is the cost of wastes, C_U is the cost of utilities, and C_L is the cost of labor. The costs of raw materials, wastes, and utilities are defined by discounting the periodic costs by a net present value factor. In the case study a net present value factor of 35,000 \$-hr is chosen to coincide with a project lifetime of five years.

The total income I_T and cost of raw materials C_{RM} are governed by the net material flows in the chemical process. Several mathematical constraints operate on these flows:

- equipment size limitations;
- cycle times for each unit process; and
- production targets.

A convenient though approximate way of imposing equipment sizing constraints is to place upper and lower bounds on the batch size of the feed to a unit process. The constraints for a unit process (i, j) take the form of Eqs. 2 and 4.

$$f_{ijk} \leq F_{ijk}^{UP}, \quad k \in K_0(i, j), \quad (2)$$

$$\text{for the current study: } F_{ijk}^{UP} [=] 1000 \text{ gal.} \quad (3)$$

$$F_{ijk}^{LO} \leq f_{ijk}, \quad k \in K_0(i, j), \quad (4)$$

$$\text{for the current study: } F_{ijk}^{LO} = 0.5 \text{ lb-mol.} \quad (5)$$

Equation six is specified by determining the molar size that corresponds to 1000 gal of feed. The cycle time constrains the production rate further by limiting the average rate at

which product can be produced for a given volume. Denoting τ_{ij} to be the nominal cycle time and δ_{ij} the increment if $x_{ij} = 1$, the averaged flowrate of the feed, \bar{f}_{ijk} , is given by Eq. 6.

$$\bar{f}_{ijk} = \frac{f_{ijk}}{\tau_{ij}} \text{ if } x_{ij} = 0 \text{ or } \frac{f_{ijk}}{\tau_{ij} + \delta_{ij}} \text{ if } x_{ij} = 1, \quad (6)$$

$$\text{for the current study: } \tau_{ij} = 24 \text{ hr; } \delta_{ij} = 3 \text{ hr } \forall (i, j). \quad (7)$$

Production targets represent lower and upper bounds on the demand for products. For many specialty and fine chemicals, there are clearly market-defined bounds on the level of production over a given time horizon. The bounds can be represented by constraints of the form in Eq. 8.

$$\sum_{(i,j) \in PS(k)} \bar{P}_{ijk} \leq \bar{P}_{ijk}^{UP}, \quad k \in K_p, \quad (8)$$

where $PS(k)$ is the set of unit processes producing stream-type k , which must be a member of the set of salable products, K_p . For the current study,

$$\bar{P}_{ijk}^{UP} [=] 200 \text{ tons year}^{-1}. \quad (9)$$

Similar bounds can be written for the minimum level of production of salable products (Eq. 10).

$$\bar{P}_{ijk}^{LO} \leq \sum_{(i,j) \in PS(k)} \bar{P}_{ijk}, \quad k \in K_p, \quad (10)$$

$$\text{for the current study: } \bar{P}_{ijk}^{LO} \equiv \frac{1}{2} \bar{P}_{ijk}^{UP}. \quad (11)$$

It is implicitly understood that one can choose not to produce a particular compound at all, in which case Eq. 10 does not apply. It should be noted that in addition to numerical bounds on the average production rate, there can also be bounds on the minimum number of products. In the present study, a nominal constraint is to produce four of the five benzonitriles at one-half or more of the production quota.

10.3.3. Unit Process Specifications

10.3.3.1. General

Unit process designs govern the internal material balances, energy balances, and equipment and utility design equations *via* specification of cycle times, split fractions, and equipment design factors (Figure 10.3).

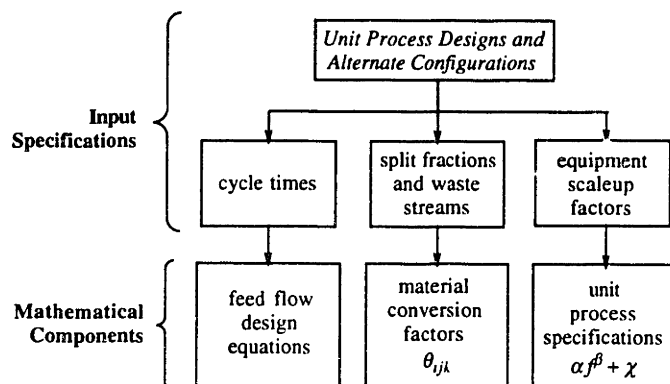


Figure 10.3. Overview of the Design Specifications Used to Parameterize Equations in the Mathematical Programming Strategy.

The design basis for each unit process is 1 lb-mole batch feed size. Designed equipment capacities are scaled proportionally to the true feed size within the mathematical programming formulation. As discussed in chapter seven, such an approach allows each of the equipment cost equations to be specified directly in terms of the batch size of the feed.

10.3.3.2. Assignment of Equipment

A set of unit processes is initially assigned to each unique production route formed by the reaction path network. Figure 10.4 depicts the set of production routes.

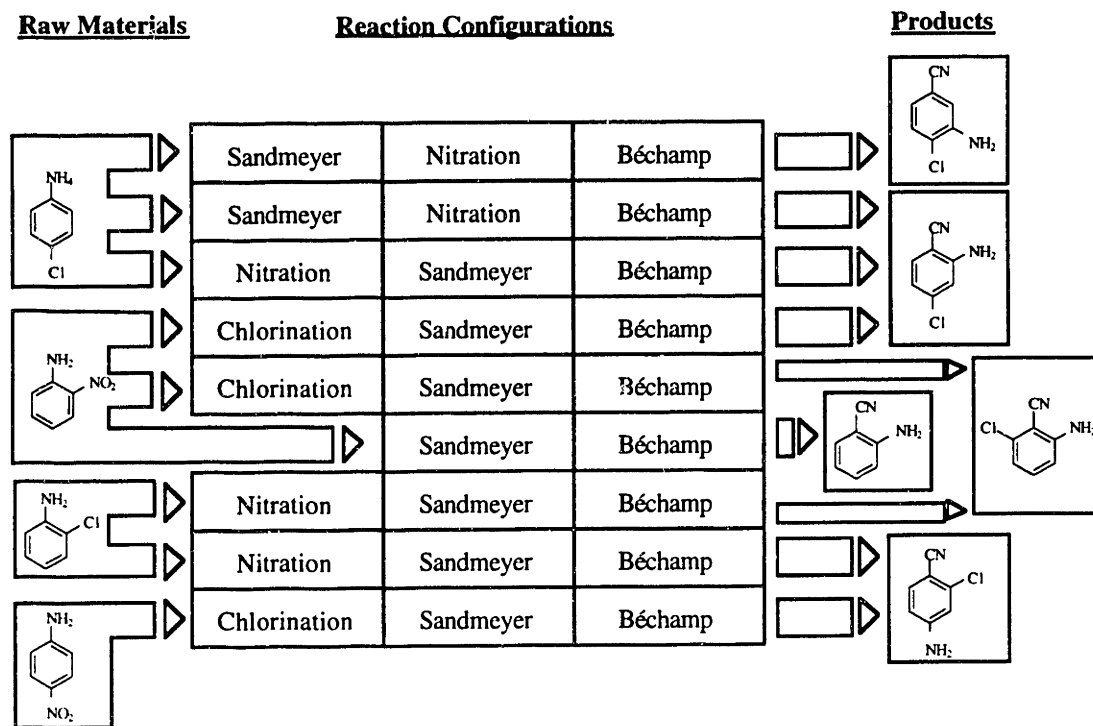


Figure 10.4. Set of Production Routes that can be Generated from the Reaction Path Network.

The first path corresponds to cyanation of 4-chloroaniline followed by nitration and reduction to yield 3-amino-4-chlorobenzonitrile. The second route proceeds with the same sequence of chemical conversions but is used to produce a different isomeric form. Remaining paths are generated in a similar fashion.

Production constraints discussed above present several opportunities to simplify Figure 10.4. The second production route involves a nitration step that yields only 1% of the main product (the by-product consumes 99% of the raw material). Almost certainly the nitration in the second path can be eliminated since its main intermediate is generated at only 1% selectivity. In addition, several of the Béchamp reductions can be merged since the magnitudes of the production quotas can be met with one unit. The use of a single large unit rather than smaller parallel units should be favored by the monotone concave shape of the cost function and the marginal level of waste generation. The resulting set of unit process alternatives is depicted in Figure 10.5.

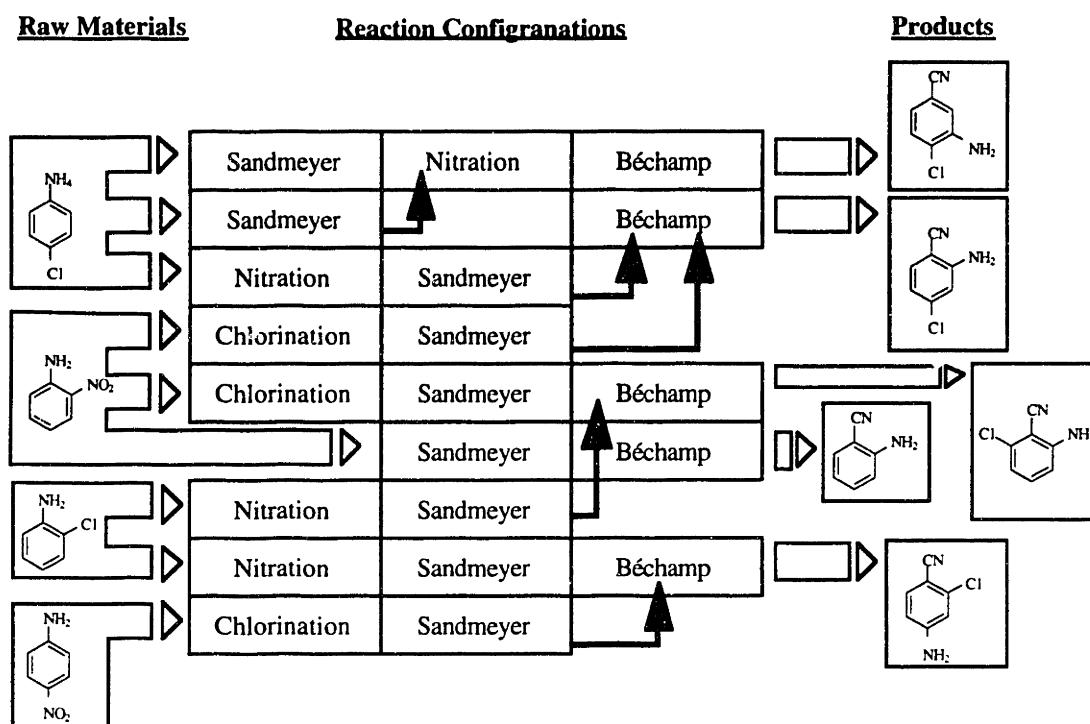


Figure 10.5. Reduced Network of Alternatives Based on the Elimination of Redundant Unit Processes. The arrows indicate the connectivity between unit processes in different reaction paths.

10.3.3.3. Equipment Design Estimates

The design estimates utilized in the study are based on

- preliminary specification of unit process flowsheets (based on available literature and patents);
- computation of material balances, energy balances, and waste streams;
- estimation of equipment capacities; and
- equipment cost estimates using power rule-type correlations.

It must be emphasized that the designs are at the study-estimate level and involve a number of approximations in the calculation of material and energy flows. Main assumptions include

- use of a lower bound in calculating material volumes and enthalpy changes;
- assuming that a vacuum still and crystallizer are adequate for separation of product mixtures; and

- and assuming a doubling of internal separator volumes for recovery of a second product.

A detailed presentation of the unit process designs is given in the supplementary section at the end of the chapter.

10.4. Results from Deterministic Formulation

10.4.1. Computational Details

Waste stream costs have been estimated from typical disposal costs and vary in the range \$0.5–1.00 per pound of material. Material prices, including raw materials, reagents, solvents, *etc.* have been obtained where possible from readily available sources such as the Chemical Market Reporter. Numerical values for waste stream costs and material prices are presented in a supplementary section at the end of the chapter.

After tightening coefficients in the mass balances to reflect maximum levels of production and after eliminating redundant variables and equations, the mathematical programming problems consist of 684 equations, 404 continuous variables, and 89 discrete variables. The problems are solved using a mixed-integer linear programming code in IBM's Optimization Subroutine Library (OSL), implemented in the General Algebraic Modeling System (GAMS) (Brooke *et al.*, 1988). Computations have been performed on a VAX 6320 computer.

10.4.2. Study 1: Effect of Intermediate Prices

A key decision in choosing a process configuration is whether to in-source the production of chloronitroaniline intermediates. Specifically, nitration and chlorination of appropriate monosubstituted anilines yield chloronitroaniline reaction by-products that can be recovered (Figure 10.1); alternately, the two conversions can be eliminated and the intermediates purchased directly. As shown in chapter nine, the isomeric mixtures are potentially amenable to separation by batch distillation. Evaluation of the unit process network, however, is still required to determine whether in-sourcing is useful in achieving the most profitable production levels. Furthermore, the sensitivity of the resulting process configuration to realistic market fluctuations should be considered to insure that a process remains profitable.

An important set of fluctuations to be considered are changes in the market prices for chloronitroaniline intermediates. In the present study, two sets of intermediate product prices are considered (Table 10.1).

Table 10.1. Instances Considered for Chloronitroaniline Market Prices.

Description	2-nitro-4-chloroaniline	2-chloro-6-nitroaniline	2-chloro-4-nitroaniline
Nominal Prices (\$ lb ⁻¹)	7.23	6.85	8.00
Reduced Prices (\$ lb ⁻¹)	5.23	4.85	4.00

At the lower market prices, one may expect the optimal process configuration to more closely resemble a set of isolated processes in which the intermediates are purchased directly. At higher prices, in-sourced production can take advantage of the spread between chloronitroaniline and monosubstituted aniline prices. Analysis of the differences in process configurations should allow several questions to be addressed: How important are synergisms and tradeoffs between reaction paths under the different price assumptions—*i.e.*, how "connected" are the optimal designs? Is there a common "superstructure-like" configuration that can be rearranged into several optimal configurations? How are the categories of income and process costs impacted?

Optimized process configurations for the two sets of intermediate prices have been obtained. A comparison of results in terms of overall categories of income and process costs is given in Table 10.2.

Table 10.2. Optimized Configurations of Chemical Reaction Paths and Aggregate Tasks. "Itns." is the number of major iterations by the solution algorithm. The units for the remaining columns are MM\$.

Description	Itns.	NPV(*)	Income	Feed	Equip.	Utilities	Waste
Nominal Prices (\$ lb ⁻¹)	592	45.13(46.27)	300.0	42.81	4.66	72.86	9.59
Reduced Prices (\$ lb ⁻¹)	998	35.41(37.55)	224.6	30.70	3.57	63.31	7.15

* Numbers in parentheses are initial linear programming relaxations.

Inspection of Table 10.2 reveals that the only category in which the first instance is favorable over the second is the total income. In the remaining categories, the second instance has lower costs across the board. At first glance, the result seems anomalous since intermediate prices are lower in the second instance. How can the income from the second process configuration possibly be smaller than the first instance if the production targets for the five benzonitriles are exactly the same? The answer can be found by examining the unit process network for the first instance (Figure 10.6). In fact, only a fraction of the produced intermediates are used within the process. The remaining quantities are sold as intermediate products. A special advantage of the approach is that the cost of producing the intermediates is presumably inelastic to its market demand due to the choice of unconventional raw materials. Furthermore, the constraints on flexibility of operation imposed by constant split fractions, common raw materials, *etc.* are dampened considerably through the sale of excess intermediate product.

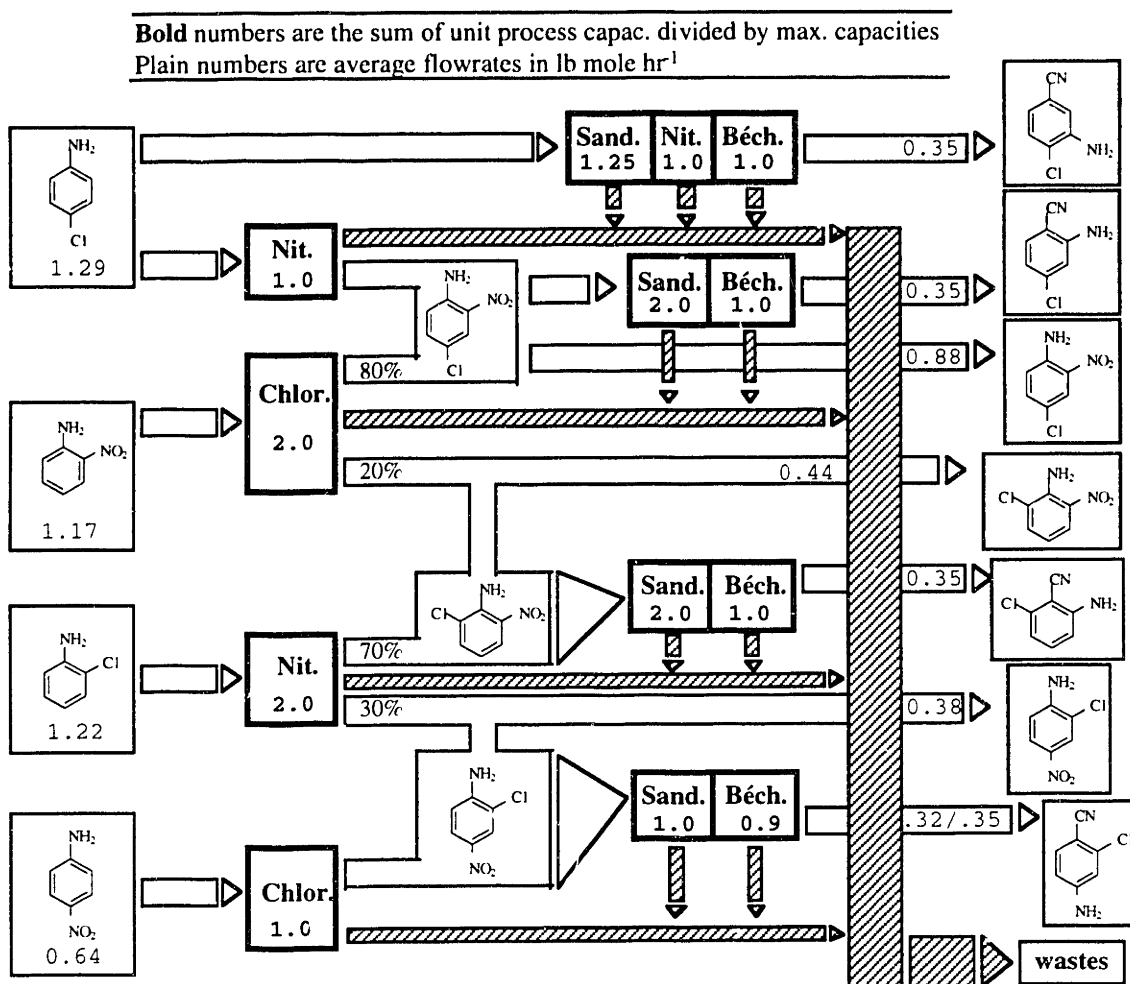


Figure 10.6. Optimal Unit Process for Nominal Intermediate Prices. Maximum production rates are shown along with chosen flow rates (separated by a forward slash) whenever the chosen rates are below maximum level.

A feature of the process depicted in Figure 10.6 is the higher level of waste stream generation as compared to the second instance (Table 10.2). Inspection of the latter process configuration (Figure 10.7), however, suggests that the key difference is that the first process produces a larger number of products. Thus, a fair comparison of the two processes in terms of overall waste generation must include the waste created in outsourced production of the intermediates. In the ammonolysis of dichloronitrobenzenes, for example, quantitative amounts of hydrogen chloride are generated which must be neutralized to sodium chloride and then disposed as a solid waste.

As expected, the second instance is closer to a traditional industrial process in that 2-chloro-4-nitroaniline is purchased directly. The resulting optimal configuration, however, still differs considerably from a set of optimal isolated processes. In particular, the split fraction for nitration of 2-chloroaniline defines the amount of makeup 2-chloro-4-

nitroaniline intermediate to be purchased. The configuration is, in fact, a subset of the configuration generated in the first instance, even though the operating strategies are very different. In short, the integrated configuration of unit processes generated by the assumption of higher intermediate prices provides a degree of flexibility not present in isolated systems. Through the choice of raw materials, intermediates, and products, a number of different configurations of the common set of unit processes can potentially be exploited to meet different design objectives and production constraints. To illustrate, the next study penalizes waste generation by giving higher weight to waste stream costs.

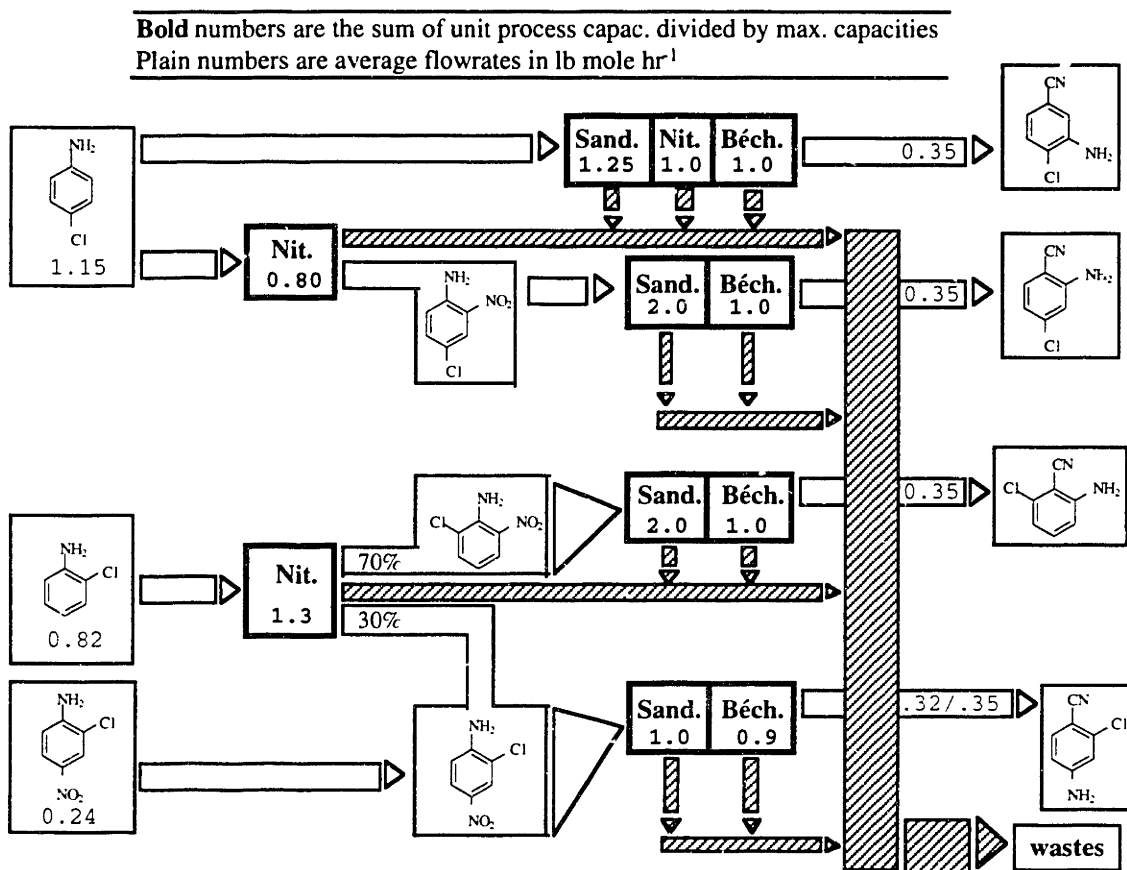


Figure 10.7. Optimal Unit Process for Reduced Intermediate Prices.

10.4.3. Study 2: Effect of Waste Stream Costs

A feature of the previous study is that optimization based purely on a net present value criterion results in higher waste stream generation within the more integrated process. The profitability of most any process, however, is sensitive to the waste treatment and disposal costs. For example, profitability of the processes depicted in the previous study could be considerably reduced over time due to higher disposal costs, tougher effluent standards (*i.e.*, additional capital equipment costs), *etc.* Such uncertainty in and of itself motivates taking steps in process development to minimize or avoid unnecessary waste generation. It is therefore desirable to determine how process integration can be exploited to actually reduce waste stream generation in the presence of production criteria that reflect environment-related goals. One approach is to optimize the unit process configuration while penalizing waste generation. This task can be accomplished most easily by placing a uniform multiplier on each of the waste costs. Due to the common set of raw materials and integrated chemical conversions present within the reaction path network, such an approach can identify processes with reduced nominal waste streams while maximizing profitability to within the assumed waste stream costs.

The specific goal of the present study is to determine how the optimal structure of the process changes when waste costs are increased by a factor of five, thus becoming a dominant component of the operating costs. The optimal unit process configuration has been obtained (Figure 10.8).

Bold numbers are the sum of unit process capac. divided by max. capacities
Plain numbers are average flowrates in lb mole hr⁻¹

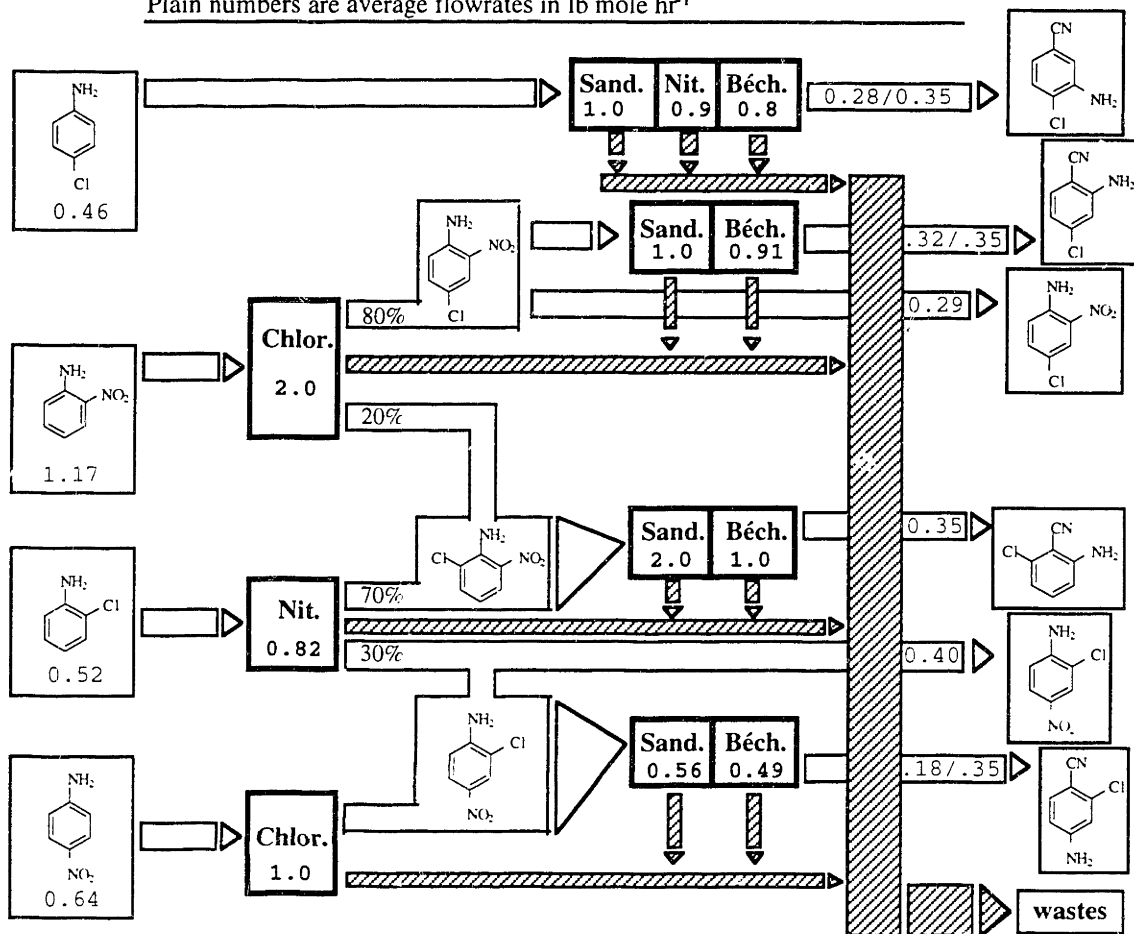


Figure 10.8. Optimal Unit Process for the Increased Waste Disposal Costs in Study 2.

The process configuration depicted in Figure 10.8 reduces the total amount of waste generated by approximately 27% on a lb-mole basis, as compared to the configuration depicted in Figure 10.8. Due to the high disposal costs of waste streams, the profitability of the process is governed by the size of the waste streams (Table 10.3). If the waste costs are held to their initial values, the evaluation function has a value of approximately 34.1 MM\$, as compared to the values from the previous study of 45.1 MM\$ and 35.4 MM\$. The details of the computational results assuming the large waste stream costs are given in Table 10.3.

Table 10.3. Optimized Configurations of Chemical Reaction Paths and Unit Processes under Different Waste Disposal Costs. "Itns." is the number of major iterations by the solution algorithm. The units for the remaining columns are MM\$.

Instance	Itns	NPV(*)	Income	Feed	Equip.	Utilities	Waste
Nominal Waste Costs	592	45.13(46.27)	300.0	42.81	4.66	72.86	9.59
Higher Waste Costs	910	11.32(13.85)	218.8	28.21	3.52	55.42	28.9

* Numbers in parentheses are initial linear programming relaxations.

The optimal production strategy is strongly dependent on tradeoffs due to production constraints, equipment capacities, and split fractions. In particular, a main way in which waste streams are reduced is through elimination of identical parallel unit processes. Such an outcome is due to the marginal increase in waste streams associated with new unit processes—*i.e.*, less incremental waste is incurred by increasing the size of a unit process than by adding a new unit process. Due to the reduced amount of equipment, the coupling between unit processes is even tighter than in the first study. For example, all of the 2-chloro-6-nitroaniline intermediate is produced internally by mixing the 20% stream from the chlorination with the 70% stream from the nitration. Although not shown in Figure 10.8, the 1% ortho by-product from nitration of 4-chlorobenzonitrile is also recovered to reduce waste generation. Hence, rather than reducing the level of process integration, the minimization of waste generation leads to a more integrated process than generated in the first study.

A basic implication of the result is that waste minimization can involve significant synergisms and tradeoffs in the choices of reaction paths. In the present study, the reaction path network can be implemented in a single multipurpose facility. In general, however, the reaction path network might involve any number of plants. Either way, regulations that target specific chemical conversions to reduce waste generation may miss the mark if they hinder relevant process integration. The most effective set of regulations should weigh all of the potential environment-related impacts in the context of a reaction path network, which includes the entire set of relevant chemistry and waste streams.

10.5. Results for the Probabilistic Formulation

Uncertainty is always present in process development. Decisions must nevertheless be made to establish priorities for the allocation of resources. In the final analysis, the research and development effort is judged by how effectively resources have been allocated—the fraction of projects undertaken that result in process implementations; and the level of resources expended on each project. It is important to recognize that process development projects can be interdependent on one another, particularly when choosing among related lines of products such as benzonitriles. The topological features of a reaction path network, in fact, translate into synergisms in the research and development effort. A knowledge of the reaction path network may suggest opportunities for integrated development of chemistry with broad applicability, choice of common raw materials, and design of transferable separations technology, to name a few.

The present section exploits the connectivity of the reaction path network to identify the most promising configuration of unit processes when there is uncertainty in the final set of products. From a process development perspective, it is important to recognize that additions or subtractions to the set of products commonly occur in design for related lines of products. Any one of a number of factors may be responsible, such as new environmental regulations, changing market analysis, or discovery of replacement products that are easier to manufacture. Such factors can occur either during or after process development. Considered here is the case where probabilistic information is available during process development. The key assumption is that such information becomes available on a short time scale as compared to the project lifetime or campaign. It is hypothesized that uncertainty in the set of products should lead to more integrated production strategies, which average the profits and costs across multiple product lines.

The chance-constrained programming strategy developed in chapter seven is employed to test the hypothesis. The existence of a product in the final product set is modeled by a random binary variable, π_k , $k \in K_f$, where $\pi_k = 1$ if the product is ultimately produced and $\pi_k = 0$ otherwise. In the present example, K_f consists of the stream-types associated with the five benzonitriles. To illustrate the generality of the representation, four of the five products are assumed to share a joint probability distribution while a fifth (3-amino-4-chlorobenzonitrile) is assumed to be independent of the others. The chance constraint for the four compounds is given by Eq. 12.

$$P \left[\pi_{k_1} \geq \frac{\bar{P}_{k_1}^{UP}}{\bar{P}_{k_1}} \text{ and } \pi_{k_2} \geq \frac{\bar{P}_{k_2}^{UP}}{\bar{P}_{k_2}} \text{ and } \pi_{k_3} \geq \frac{\bar{P}_{k_3}^{UP}}{\bar{P}_{k_3}} \text{ and } \pi_{k_4} \geq \frac{\bar{P}_{k_4}^{UP}}{\bar{P}_{k_4}} \right] \geq a_1, \quad (12)$$

where $k_1, k_2, k_3, k_4 \in K_f$ are the stream-type indices for 2-amino-4-chlorobenzonitrile, 2-aminobenzonitrile, 2-chloro-6-aminobenzonitrile, and 2-chloro-4-aminobenzonitrile, \bar{P}_k is

the net production rate, and \bar{P}_k^{UP} is an upper bound on the production rate. The probabilities for each of the sixteen possible outcomes are given in Table 10.4.

Table 10.4. Candidate Total Production Objectives and Associated Probabilities for a Set of Four Benzotrioles.

Outcome: $e \in E$	$\bar{\pi}_k \in \{0, 1\}, k \in K_f$				Prob. Outcome: ρ_e
	2-NH ₂ -4-Cl-	2-NH ₂ -	2-Cl-6-NH ₂ -	2-Cl-4-NH ₂ -	
1	1	1	1	1	0.354
2	1	1	1	0	0.093
3	1	1	0	1	0.093
4	1	1	0	0	0.031
5	1	0	1	1	0.093
6	1	0	1	0	0.031
7	1	0	0	1	0.031
8	1	0	0	0	0.011
9	0	1	1	1	0.093
10	0	1	1	0	0.031
11	0	1	0	1	0.031
12	0	1	0	0	0.011
13	0	0	1	1	0.031
14	0	0	1	0	0.011
15	0	0	0	1	0.011
16	0	0	0	0	0.044

Uncertainty in the production of 3-amino-4-chlorobenzonitrile is assumed to be 25%—i.e., there is a 25% probability that 3-amino-4-chlorobenzonitrile will not be produced. The chance constraint for 3-amino-4-chlorobenzonitrile is written as

$$P \left[\pi_{k_5} \geq \frac{\bar{P}_{k_5}^{UP}}{\bar{P}_{k_5}} \right] \geq a_2. \tag{13}$$

The objective function can also be recast in a probabilistic context. The objective function in the deterministic formulation is given by

$$\max NPV,$$

where NPV is a metric of the net present value (Eq. 1). The introduction of a probability distribution associated with the production target constraints motivates the formulation of a new objective function. The chosen form of the objective function is given by Eq. 14.

$$\max \left\{ \left(\frac{1}{\text{card}(E)} \sum_{e \in E} \rho_e NPV_e \right) + W_1 a_1 + W_2 a_2 \right\} \tag{14}$$

The first term in the objective function is the expectation of the net present value averaged over all of the possible outcomes. The net present value metric is modified from the deterministic formulation by adjusting the cost coefficients for the products. Specifically, for an event $e \in E$,

$$Price_k^e(\text{probabilistic}) \leftarrow Price_k(\text{deterministic}) \text{ if } \tilde{\pi}_k = 1, k \in K_p \quad (15)$$

$$Price_k^e(\text{probabilistic}) \leftarrow \text{waste disposal cost} \text{ if } \tilde{\pi}_k = 0, k \in K_p. \quad (16)$$

Equations 15 and 16 are the only modifications to the deterministic form of the net present value formula.

The terms $W_1 a_1$ and $W_2 a_2$ are penalty terms associated with the chance-constraints. These terms can be understood by categorizing the types of designs that can be created. When the penalty terms are small, the resulting process configurations tend to maximize the expected value of the configuration. It is important to realize, however, that a significant variance may be associated with the expected value—*i.e.*, any realization of the process configuration may have a net present value that is considerably different than the average expected value. When the penalty terms are dominant, the process configuration attempts to maximize the probability of realizing the final choice of products in the actual process. Comparison of the two regimes may indicate that one is more important or useful than the other in selecting a flexible unit process configurations.

To simplify the analysis, W_1 and W_2 are related by a proportionality constant. The constant is chosen as a scaling factor to reflect the fact that a_2 can vary over a considerably larger range than a_1 . The resulting form of the weighting function incorporated in the calculations is given by Eq. 17.

$$W_1 a_1 + W_2 a_2 = W \cdot (a_1 + 0.0147 a_2) \quad (17)$$

An additional constraint on the problem is that at least one of the five main products must be produced. The constraint effectively relaxes the deterministic constraint that at least four of the five products must be produced. When the optimizations are performed, an interesting parametric dependence on W is observed. Three distinct ranges of W appear which yield single process configurations and expected net present values throughout each range. The ranges are W *ca.* 0–40, 40–60 and 60–100. As illustrated in Table 10.5, increasing values for the weighting functions lead to a reduction in the expected net present value and increases in a_1 and a_2 .

Table 10.5. Chance-Constrained Configurations of Chemical Reaction Paths and Unit Processes. The unit for the net present values is MM\$.

Range of W	Objective Func.	$E[NPV]^*$	a_1	a_2
0.00–13.75		17.46	0.011	0.25
17.50–50.00	17.72–19.51	16.76	0.044	0.75
56.25–100.00	21.2–36.84	1.48	0.354	0.25

* Expectation value

The three designs differ considerably from one another in their topological structure. The designs for the ranges 10–13.75 and 56.25–100 are depicted in Figures 10.9 and 10.10.

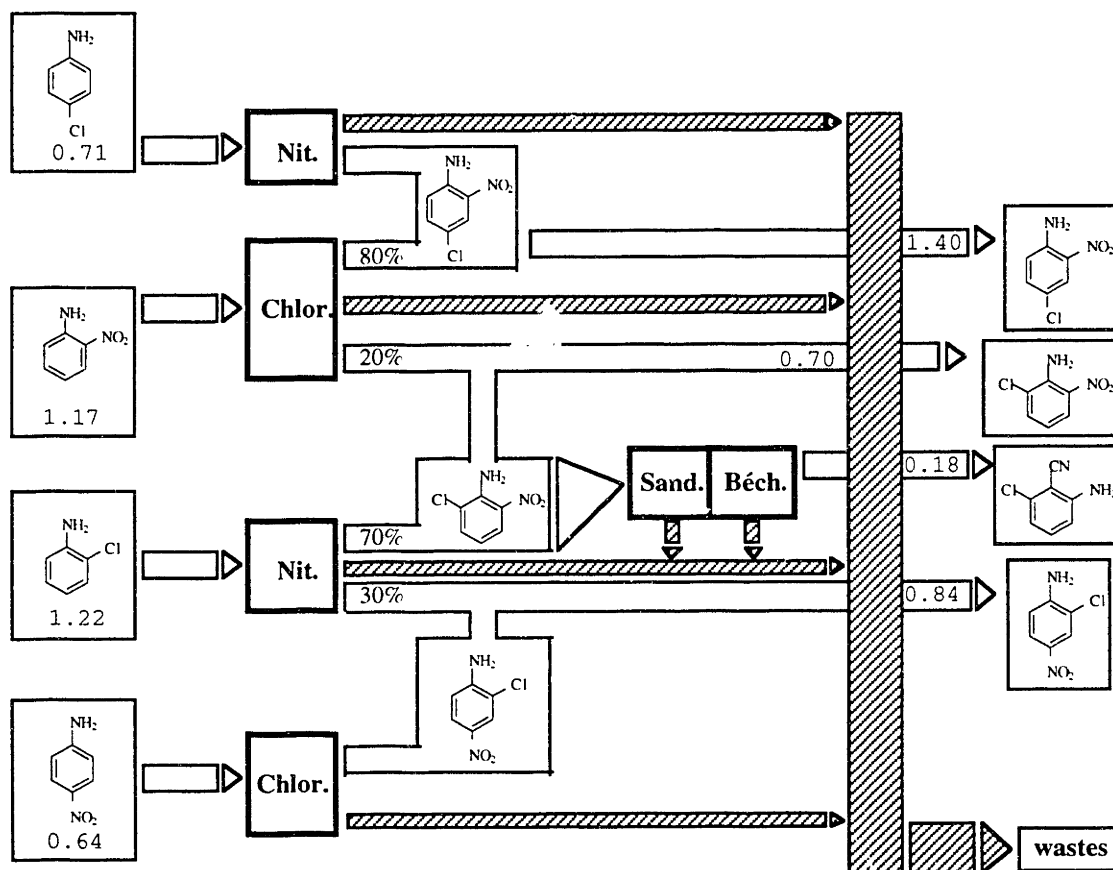


Figure 10.9. Computational Results for the Chance-Constrained Programming Formulation with W in the range 10–13.75.

Figure 10.9 presents the process configuration that most directly maximizes the net present value. Since the penalty contribution is relatively small, one might expect the configuration to most resemble the deterministic cases. To the contrary, the expectation value criterion in fact shifts production away from the target benzonitriles and towards the

intermediates, for which no production risk is specified. Hence, the expectation value criterion does not seem to lead to an averaging of risk under the production requirements of the problem. For weighting factors in the numerical range >56.25 , however, the expectation value becomes a relatively small contribution to the objective function. The design objective is dominated by the chance-constrained penalty terms, which place an emphasis on maximizing the realization probability of a particular configuration. Under the chance-constrained criterion the resulting design configuration takes a form that most directly spreads the risk of production by combining a number of inter-related production routes. Thus, the chance-constrained approach yields designs which most closely resemble the deterministic results from the previous section.

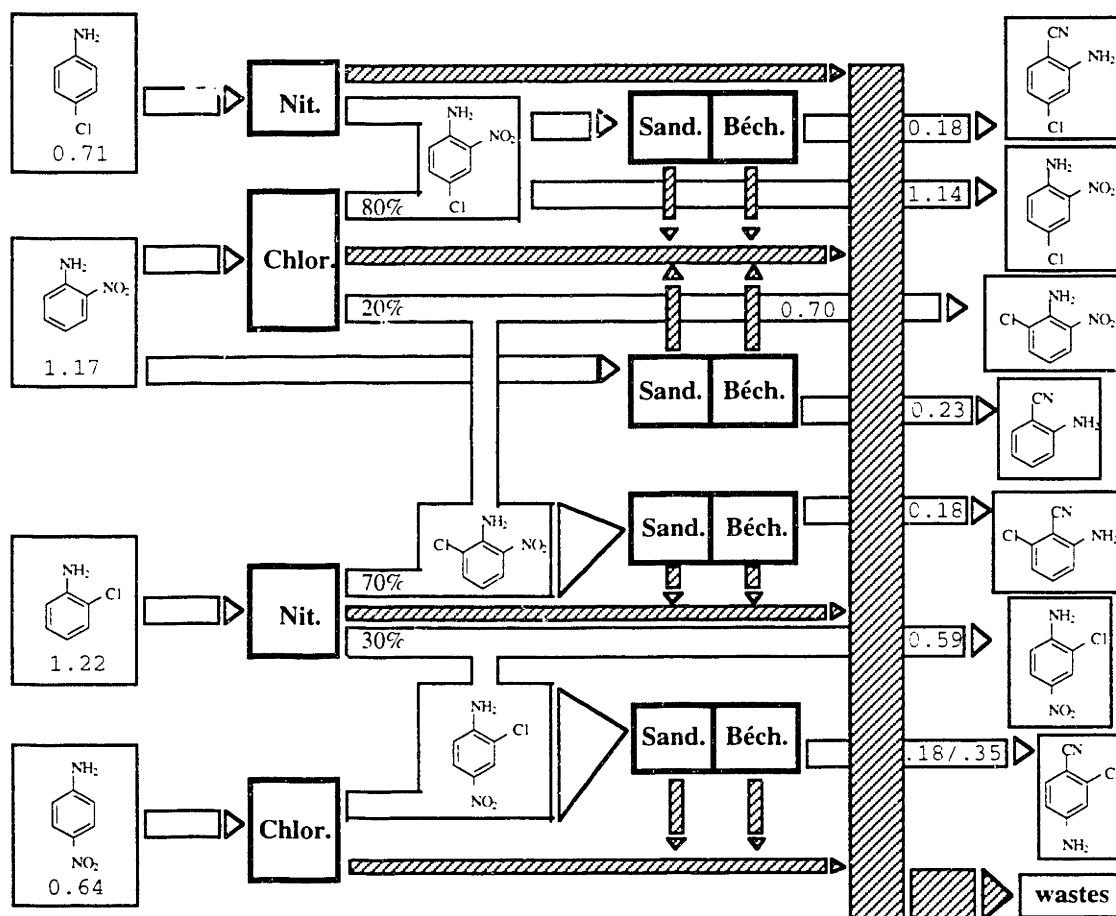


Figure 10.10. Computational Results for the Chance-Constrained Programming Formulation with W in the range 56.25-1000.

While the probabilistic formulation does not directly account for the set of operating responses that can be generated from changes in production goals, several implications can be inferred from the proposed configurations. A configuration such as shown in Figure 10.10, for example, could potentially respond to changes in the set of products by

deploying the set of unit processes to maximize the actual net present value. By contrast, the configuration depicted in Figure 10.9 can hardly be considered for the production of significant quantities of the target benzonitriles. Hence, it is concluded that the chance-constrained penalty function is a promising strategy for the creation of designs which distribute risk though the integration of chemical production routes.

10.6. Discussion and Extensions

The general theme pursued in the present chapter has been the importance of considering multiple products in preliminary process screening. A highly integrated process configuration has been obtained for maximization of a net present value metric. When waste streams costs are significantly increased, it is then found that the integration of the network becomes even tighter due to the reduction in redundant equipment. The two results suggest that reaction path networks should be considered under both economic and environment-related design objectives, as well as objectives that attempt to incorporate both. It is important to recognize that development of an integrated production strategy can allow both a reduction in environmental waste and an increase in profitability in comparison to existing non-integrated production routes. This possibility should be recognized in environment-related regulations and carefully considered before attempting to regulate specific chemical conversions and compounds that are part of a larger reaction path network. In other words, the net environment-related impact of removing or modifying a chemical conversion from a reaction path network should be considered.

Process flexibility is traditionally measured by the ability to respond to changes in operating parameters through adjustment of unit operation control variables. Consideration of the synergisms between reaction paths, in terms of the underlying choice of raw materials, intermediates, and chemical conversions, leads to the observation that process chemistry can lead to an additional dimension of flexibility. For example, the ability to perform (or not) additional separations to recover a usable by-product enables a response to higher waste disposal costs. The ability to change the product line by rescheduling a common set of unit processes, manufacturing an intermediate product in excess for sale, etc. can all be considered control variables derived from the choice of chemistry. Such options have always been an implicit consideration in the development of chemistry. The new design tools, however, allow a more complete and larger scale consideration of the potential design implications. The type of flexibility associated with the choice of chemistry is also relevant to process development. Choice of common chemical conversions to several promising products, for example, results in a reduced development risk. Furthermore, an explicit accounting for the uncertainty in future production targets exploits reaction path integration to average risk associated with uncertain production targets.

A third implication of the results is that the most important areas for more detailed process modeling can be identified as those which are most critical to the proposed process

configurations. For example, the feasibility of separations and the design of unit processes for in-sourced production of chloronitroaniline intermediates is critical to most of the proposed reaction path networks. Chapter nine has illustrated how one might go about such analysis early-on in process development. A logical next step in computer-aided analysis would be to develop converged process flowsheet simulations of the production routes. Resulting economic analysis of the flowsheets represents improved model parameters and could be reincorporated into the total plant synthesis formulation for more accurate and refined analysis. Dynamic simulation tools (appendix three) can be employed to obtain better estimates of cycle times for the unit processes. Finally, scheduling tools based on similar mathematical optimization techniques can be used to determine the feasibility of implementing the integrated process configurations in a multipurpose plant.

10.7. Conclusions

The total plant synthesis analysis of the reaction path network has been employed to better understand the process-level attributes of the chemical reaction paths. It is concluded that a consideration of multiple products allows improved or optimal process configurations to be obtained by exploiting synergism and tradeoffs between the underlying chemical reaction paths. Furthermore, availability of multipurpose raw materials and chemical conversions provides a basis both for operating flexibility and reduction of risk in process development. One of the implications of considering multiple products is that environment-related regulations should take into account the relevant reaction path network, which includes all of the relevant chemistry and waste streams.

10.8. Supplement: Unit Process Specifications

10.8.1. Nitration

As discussed in chapter eight, aromatic nitration consists of the replacement of a hydrogen on the aromatic nucleus by a nitro group. The overall stoichiometry can be broken into the several reactions shown in Table 10.6.

Table 10.6. Reaction Stoichiometry for Nitration in Mixed Acetic Acid / Acetic Anhydride.

<i>R1: Protection of Amino Group</i>	$\text{H-Ar-NH}_2 + (\text{CH}_3\text{CO})_2\text{O} \rightarrow \text{H-Ar-NHCOCH}_3 + \text{CH}_3\text{COOH}$
<i>R2: Main Reaction</i>	$\text{H-Ar-NHCOCH}_3 + \text{HNO}_3 + (\text{CH}_3\text{CO})_2\text{O} \rightarrow \text{O}_2\text{N-Ar-NHCOCH}_3 + 2 \text{CH}_3\text{COOH}$
<i>R3: Recovery of Amino Group</i>	$\text{O}_2\text{N-Ar-NHCOCH}_3 + \text{H}_2\text{O} \rightarrow \text{O}_2\text{N-Ar-NH}_2 + \text{CH}_3\text{COOH}$
<i>Net Stoichiometry</i>	$\text{H-Ar-NH}_2 + 2 (\text{CH}_3\text{CO})_2\text{O} + \text{HNO}_3 + \text{H}_2\text{O} \rightarrow \text{O}_2\text{N-Ar-NH}_2 + 4 \text{CH}_3\text{COOH}$

Amino substituents must usually be protected from the nitronium-ion related electrophile, which can cause oxidation of the amino group. Certain aniline derivatives (*i.e.*, containing an amino group), in fact, form explosive oxidation products upon addition of nitration reagents. An often used approach for anilines and phenols is to first convert the aniline to an acetanilide (*i.e.*, the amino group is replaced by NHCOCH_3) intermediate by reaction with acetic anhydride. Based on available literature, the remaining substituents (*i.e.*, Cl, CN, and NO_2) do not appear to cause significant competing reactions with the desired nitration in acetic anhydride or acetic acid. The amidation and hydrolysis reactions, *R1* and *R4*, can be treated separately from the main reaction, *R2*. Reactions *R1* and *R4* are assumed to yield essentially complete conversion.

The general process elements in the nitration unit process are reported in several places, including Groggins (1952), although they are almost always for nitration in mixed-acid media. A useful feature of reaction in acetic anhydride is the capability to perform the functional group chemistry within the reactor and deprotection within the neutralization stage. Separation consists of two main trains. Initial separation of product(s) from the spent reagent is accomplished by cooling and crystallization of the neutralization product. The compound class under consideration is at most sparingly soluble in acetic anhydride / acetic acid. If necessary, the solution can be pre-concentrated by evaporation (*e.g.*, using thin film evaporation) if necessary. The concentration of acetic anhydride can be reduced if necessary by hydrolysis. Purification of the isolated products is accomplished by vacuum distillation. Additionally, crystallization can be utilized if necessary. The resulting process configuration is depicted in Figure 10.11.

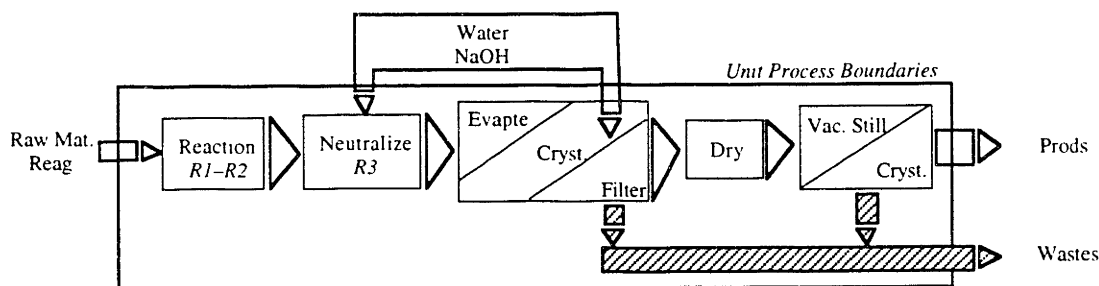


Figure 10.11. Rough Sketch of a Unit Process for Homogeneous Nitration.

Short cut design estimates for the individual unit operations in Figure 10.11 are given in Table 10.7.

Table 10.7. Summary of Equipment and Utility Costing Equations for Nitration.

Equipment Item:	Utilities	Sizing Equation	Pred. Capacity	Unit Process Eq.	Utilities
<i>Single Product</i>					
reactor:	—	$2.28 \times 10^4 (\text{gal}/300)^{0.48}$	58.4 gal	$1.1 \times 10^4 f^{0.48}$	—
neutralizer:	—	$0.22 \times 10^4 (\text{gal}/100)^{0.52}$	65.0 gal	$0.18 \times 10^4 f^{0.52}$	—
evaporator:	steam	$2.1 \times 10^4 (\text{ft}^2/1000)^{0.62}$	500 ft ²	$1.4 \times 10^4 f^{0.62}$	$0.18 \times 10^4 \bar{f}$
dryer:	steam	$51.8 \times 10^4 (m^3 + 0.41)$	0.11 m ³	$5.7 \times 10^4 (6f + 3.7)$	$0.87 \times 10^4 \bar{f}$
storage (2 vess.):	—	$0.1 \times 10^4 (\text{gal}/100)^{0.52}$	25.8 gal	$0.1 \times 10^4 f^{0.52}$	—
still:	n.g., steam	$2.28 \times 10^4 (\text{gal}/300)^{0.48}$	14.7 gal	$0.53 \times 10^4 f^{0.48}$	$6.85 \times 10^4 \bar{f}$
collection vess.(3):	—	$0.1 \times 10^4 (\text{gal}/100)^{0.52}$	14.7 gal	$0.12 \times 10^4 f^{0.52}$	—
crystallizer:	—	$3.0 \times 10^4 (\text{gal}/500)^{0.26}$	—	—	—
<i>Recovery of Second Product (if there is one)</i>					
		new size of still:	30.0 gal	$0.75 \times 10^4 f^{0.48}$	
		new size of collection vessels:	30.0 gal	$0.17 \times 10^4 f^{0.52}$	
		new still utilities:			$13.7 \times 10^4 \bar{f}$
		new size of crystallizer:	30.0 gal	$1.5 \times 10^4 f^{0.26}$	
<i>Additional Costs</i>					
			materials	utilities	misc.
		reagents:	142.0 f		
		labor:			$12.5 f^{0.25}$
		electricity:		$1.14 \bar{f}$	

References and Comments on Table 10.7. P&T refers to Peters and Timmerhaus (1980); P&G refers to Perry and Green (1984). **Reactor:** glass-lined (cost factor of 1.2), jacketed iron kettle; P&T, pg. 790 installed cost for January, 1979. **Neutralizer:** mixing tank with agitator; P&T, pg. 572 for January, 1979. **Evaporator:** falling film evaporator with condenser (cost factor of 1.2); P&G, pg. 11-42 installed cost for mid-1968. **Dryer:** rotary vacuum dryer; P&G, pg. 20-44 assumed for January, 1979. **Storage Tank:** carbon steel storage tank; P&T, pg. 572 for January, 1979. **Distillation Still:** glass lined vessel with furnace and condenser units (cost factor of 1.2); P&T, pg. 790 installed cost for January, 1979. **Collection Vessels:** see Storage Tank. **Crystallizer:** vacuum-batch crystallizer; P&T, pg. 593 installed cost for January, 1979. Periodic costs such as utilities, materials, etc. converted to a net present value using a multiplier of 35,000 \$-hr, which corresponds to a

project lifetime of five years. **Labor costs:** operation of multiple small batch units; P&T, pg. 196.

10.8.2. Chlorination

Chlorination of protected nitroaniline substrates occurs readily by molecular chlorine in acetic acid solvent. Chlorination is an electrophilic aromatic substitution reaction. The process chemistry can be represented by three basic chemical conversions (Table 10.8).

Table 10.8. Reaction Stoichiometry for Chlorination in Acetic Acid.

<i>R1: Protection of Amino Group</i>	$\text{H-Ar-NH}_2 + (\text{CH}_3\text{CO})_2\text{O} \rightarrow \text{H-Ar-NHCOCH}_3 + \text{CH}_3\text{COOH}$
<i>R2: Main Reaction</i>	$\text{H-Ar-NHCOCH}_3 + \text{Cl}_2 \rightarrow \text{Cl-Ar-NHCOCH}_3 + \text{HCl}$
<i>R3: Recovery of Amino Group</i>	$\text{Cl-Ar-NHCOCH}_3 + \text{H}_2\text{O} \rightarrow \text{Cl-Ar-NH}_2 + \text{CH}_3\text{COOH}$
<i>Net Stoichiometry</i>	$\text{H-Ar-NH}_2 + (\text{CH}_3\text{CO})_2\text{O} + \text{Cl}_2 + \text{H}_2\text{O} \rightarrow \text{Cl-Ar-NH}_2 + \text{HCl} + 2 \text{CH}_3\text{COOH}$

Conversion of the amino group to the amide by reaction with acetic anhydride is desirable to reduce the activation of the substrate, which can lead to polychlorination. Examination of the partial rate factors for chlorination (chapter eight) shows that, unlike nitration, the chlorination reaction tends to be subject to multiple substitution due to the amide group. Not only is the partial rate factor for the amide group larger, but the chlorine atom is insufficiently deactivating to prevent further substitution. Thus, the problem is an instance of a general polysubstitution transform network.

Chlorination of nitroanilines has been the subject of considerable patent literature (*e.g.*, Smyth *et al.*, 1991). The use of acetic acid is desirable because it is a benign solvent and one that can be recovered by crystallization of the reaction product. The chlorination reaction generally is carried out in a gas-sparged reactor and the reaction products readily separated from the solvent by crystallization, with or without preliminary concentration of the product in a thin film evaporator. The products can be dried together and then distilled, crystallized, and recrystallized to achieve the desired product purity. It has been shown in chapter nine that the product mixtures may be well suited to vacuum distillation. As a result, the separations portion of the process flowsheets for nitration and chlorination are similar. Estimated unit process capacities and costs are given in Table 10.9.

Table 10.9. Summary of Equipment and Utility Costing Equations for Chlorination.

Equipment Item:	Utilities	Sizing Equation	Pred. Capacity	Unit Process Eq.	Utilities
<i>Single Product</i>					
reactor:	—	$1.9 \times 10^4 (\text{gal}/300)^{0.48}$	55.5 gal	$0.85 \times 10^4 f^{0.48}$	—
neutralizer:	—	$0.22 \times 10^4 (\text{gal}/100)^{0.52}$	63.4 gal	$0.17 \times 10^4 f^{0.52}$	—
settling vessel:	—	$0.1 \times 10^4 (\text{gal}/100)^{0.52}$	63.4 gal	$0.08 \times 10^4 f^{0.52}$	—
evaporator:	steam	$2.1 \times 10^4 (\text{ft}^2/1000)^{0.62}$	500 ft ²	$1.4 \times 10^4 f^{0.62}$	$0.24 \times 10^4 \bar{f}$
dryer:	steam	$51.8 \times 10^4 (m^3 + 0.41)$	0.11 m ³	$5.7 \times 10^4 (f + 3.7)$	$0.37 \times 10^4 \bar{f}$
storage (2):	—	$0.1 \times 10^4 (\text{gal}/100)^{0.52}$	29.0 gal	$0.11 \times 10^4 f^{0.52}$	—
still:	n.g., steam	$2.28 \times 10^4 (\text{gal}/300)^{0.48}$	12.0 gal	$0.49 \times 10^4 f^{0.48}$	$5.61 \times 10^4 \bar{f}$
collection vess.(3):	—	$0.1 \times 10^4 (\text{gal}/100)^{0.52}$	12.0 gal	$0.1 \times 10^4 f^{0.52}$	—
crystallizer:	—	$3.0 \times 10^4 (\text{gal}/500)^{0.26}$	—	—	—
<i>Recovery of Second Product (if there is one)</i>					
			new size of still:	24.0 gal	$0.67 \times 10^4 f^{0.48}$
			new size of collection vessels:	24.0 gal	$0.15 \times 10^4 f^{0.52}$
			new still utilities:		$11.2 \times 10^4 \bar{f}$
			new size of crystallizer:	24.0 gal	$1.35 \times 10^4 f^{0.26}$
<i>Additional Costs</i>					
			materials	utilities	misc.
			reagents:	141.2 f	
			labor:		$12.5 f^{0.25}$
			electricity:	1.14 \bar{f}	

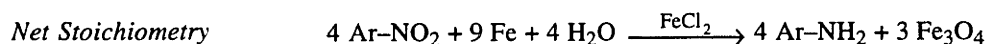
References and Comments on Table 10.9. **P&T** refers to Peters and Timmerhaus (1980); **P&G** refers to Perry and Green (1984). **Reactor:** jacketed iron kettle; **P&T**, pg. 790 installed cost for January, 1979. **Neutralizer:** mixing tank with agitator; **P&T**, pg. 572 for January, 1979. **Settling Tank:** carbon steel storage tank; **P&T**, pg. 572 for January, 1979. **Evaporator:** falling film evaporator with condenser (cost factor of 1.2); **P&G**, pg. 11-42 installed cost for mid-1968. **Dryer:** rotary vacuum dryer; **P&G**, pg. 20-44 assumed for January, 1979. **Storage Tank:** carbon steel storage tank; **P&T**, pg. 572 for January, 1979. **Distillation Still:** glass lined vessel with furnace and condenser units (cost factor of 1.2); **P&T**, pg. 790 installed cost for January, 1979. **Collection Vessels:** see Storage Tank. **Crystallizer:**

vacuum-batch crystallizer; P&T, pg. 593 installed cost for January, 1979. Periodic costs such as utilities, materials, *etc.* converted to a net present value using a multiplier of 35,000 \$-hr, which corresponds to a project lifetime of five years. **Labor costs:** operation of multiple small batch units; P&T, pg. 196.

10.8.3. Béchamp Reduction

The Béchamp reduction is a classic chemical process and one which is widely used on substrates of the type considered in this case study. The main reaction is a complicated one but can be summarized as the oxidation of iron in acid media to affect the reduction of the amino group (Table 10.10). The ferrous chloride catalyst, FeCl_2 , is generated in much larger than catalytic amounts by reaction between HCl and iron filings.

Table 10.10. Reaction Stoichiometry for the Béchamp Reduction.



A detailed description of the Béchamp process is given by Shreve (1963). In brief, the reaction is carried out in two reaction vessels, one for the main reduction and one for supplying reaction product to a filter press, where a dilute aqueous solution containing the product is separated from iron oxide by-product. The remaining components of the process consist of three main steps: (i) filtration of the reduction batch to remove the iron sludge; (ii) evaporation and drying of the reaction product; and (iii) vacuum distillation of the reaction product. A diagram of the tasks is depicted in Figure 10.12.

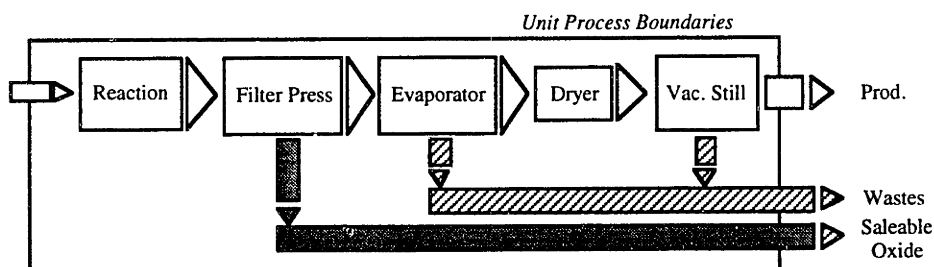


Figure 10.12. Rough Sketch of the Béchamp Process for Nitrobenzotrioles.

Estimated unit process capacities and costs are given in Table 10.11.

Table 10.11. Summary of Equipment and Utility Costing Equations for the Béchamp Reduction.

Equipment Item:	Utilities	Sizing Equation	Pred Capacity	Unit Process Eq.	Utilities
-----------------	-----------	-----------------	---------------	------------------	-----------

		<u>Single Product</u>			
reactor(2):	—	$1.9 \times 10^4 (\text{gal}/300)^{0.48}$	68.0 gal	$2.4 \times 10^4 f^{0.48}$	—
diaphragm pump:	—	$0.3 \times 10^4 (\text{gal hr}^{-1}/1800)^{0.29}$	408 gal hr ⁻¹	$0.19 \times 10^4 f^{0.29}$	—
filter press:	—	$0.4 \times 10^4 (\text{ft}^2/90)^{0.34}$	100 ft ²	$0.42 \times 10^4 f^{0.34}$	—
filtrate storage:	—	$0.1 \times 10^4 (\text{gal}/100)^{0.52}$	96.5 gal	$0.1 \times 10^4 f^{0.52}$	—
evaporator:	steam	$2.1 \times 10^4 (\text{ft}^2/1000)^{0.62}$	2000 ft ²	$3.7 \times 10^4 f^{0.62}$	$5.1 \times 10^4 \bar{f}$
dryer:	steam	$51.8 \times 10^4 (m^3 + 0.41)$	0.11 m ³	$5.7 \times 10^4 (f + 3.7)$	$1.03 \times 10^4 \bar{f}$
storage (2):	—	$0.1 \times 10^4 (\text{gal}/100)^{0.52}$	82.0 gal	$0.18 \times 10^4 f^{0.52}$	—
still:	n.g., steam	$2.28 \times 10^4 (\text{gal}/300)^{0.48}$	11.0 gal	$0.46 \times 10^4 f^{0.48}$	$5.0 \times 10^4 \bar{f}$
collection vess.(2):	—	$0.1 \times 10^4 (\text{gal}/100)^{0.52}$	11.0 gal	$0.07 \times 10^4 f^{0.52}$	—
<u>Additional Costs</u>					
			materials	utilities	misc.
	reagents:		25.0 f		
	labor:				12.5 f ^{0.25}
	electricity:			3.7 \bar{f}	

References and Comments on Table 10.11. P&T refers to Peters and Timmerhaus (1980); P&G refers to Perry and Green (1984). **Reactor:** enameled steel kettle (cost factor of 1.2 as jacketed iron kettle); P&T, pg. 790 installed cost for January, 1979. **Diaphragm Pump:** P&T, pg. 558 for January, 1979. **Filter Press:** P&T, pg. 589 for January, 1979. **Filtrate Storage:** carbon steel storage tank; P&T, pg. 572 for January, 1979. **Evaporator:** falling film evaporator with condenser (cost factor of 1.2); P&G, pg. 11-42 installed cost for mid-1968. **Dryer:** rotary vacuum dryer; P&G, pg. 20-44 assumed for January, 1979. **Storage Tank:** carbon steel storage tank; P&T, pg. 572 for January, 1979. **Distillation Still:** glass lined vessel with furnace and condenser units (cost factor of 1.2); P&T, pg. 790 installed cost for January, 1979. **Collection Vessels:** see Storage Tank. Periodic costs such as utilities, materials, etc. converted to a net present value using a multiplier of 35,000 \$-hr, which corresponds to a project lifetime of five years. **Labor costs:** operation of multiple small batch units; P&T, pg. 196.

10.8.4. Sandmeyer Reaction for Nitrile Synthesis

As discussed in chapter eight, the Sandmeyer conversion requires two main conversions. First, it consists of the diazotization of an aromatic amine to form a diazonium salt intermediate. This intermediate, while still in solution, then undergoes catalyzed free

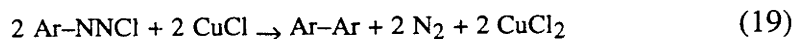
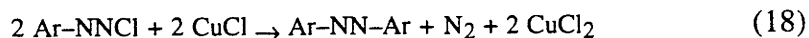
radical addition leading to the formation of the corresponding benzonitrile. The two main elements of the reaction stoichiometry are given in Table 10.12.

Table 10.12. Reaction Stoichiometry for Sandmeyer Reaction.

<i>Formation of Diazot. Reagent</i>	$\text{NaNO}_2 + \text{HCl} \rightarrow \text{HONO} + \text{NaCl}$
<i>Diazotization</i>	$\text{Ar-NH}_2 + \text{HONO} + \text{HCl} \rightarrow \text{Ar-NNCl} + 2 \text{H}_2\text{O}$
<i>R1: Net Diazot. Stoichiometry</i>	$\text{Ar-NH}_2 + \text{NaNO}_2 + 2 \text{HCl} \rightarrow \text{Ar-NNCl} + 2 \text{H}_2\text{O}$
<i>Formation of Cu Cyanide Cat.</i>	$2 \text{KCN} + \text{Cu}_2(\text{CN})_2 \rightarrow 2 \text{KCu}(\text{CN})_2$
<i>Sandmeyer Reaction</i>	$\text{Ar-NNCl} + \text{KCu}(\text{CN})_2 \rightarrow \text{Ar-CN} + \text{N}_2 + \text{KCl} + 0.5 \text{Cu}_2(\text{CN})_2$
<i>R2: Net Sand. Rxn. Stoich.</i>	$\text{Ar-NNCl} + \text{KCN} \xrightarrow{\text{Cu}_2(\text{CN})_2 / \text{ca. } 5^\circ\text{C}} \text{Ar-CN} + \text{N}_2 + \text{KCl}$
<i>Net Stoichiometry</i>	$\text{Ar-NH}_2 + \text{NaNO}_2 + 2 \text{HCl} + \text{KCN} \xrightarrow{\text{Cu}_2(\text{CN})_2} \text{Ar-CN} + \text{N}_2 + \text{KCl} + 2 \text{H}_2\text{O}$

The generation of the diazonium salt almost always accompanies the Sandmeyer reaction. Thus, the chemical transforms defined for the Sandmeyer reaction should start with the amino group rather than the diazonium salt. The diazonium salt is usually synthesized in an acidified solution of sodium nitrite according to the stoichiometry given in Table 10.

As discussed in chapter two, the diazotization and Sandmeyer reactions are not described well by linear free-energy relationships. For the compounds under consideration, however, many results (Hagedorn, 1985) show that the reactions can be assumed to proceed rapidly and to consume essentially all of the reactants. The major issue with these compounds is the formation of by-products due to competing transforms. For example, by-products can result from copper-catalyzed diazo coupling reactions. Two commonly encountered side reactions are given in Eqs. 18 and 19.



Following the reaction steps discussed above, the reaction product is crystallized, dried, and sent to a vacuum still to separate the reaction product from heavier by-products (Figure 10.13). It is important to recognize that additional equipment is required to remove any hydrogen cyanide from the off-gas free and liquid-phase reaction by-products any hydrogen cyanide that has been generated (Hagedorn, 1985).

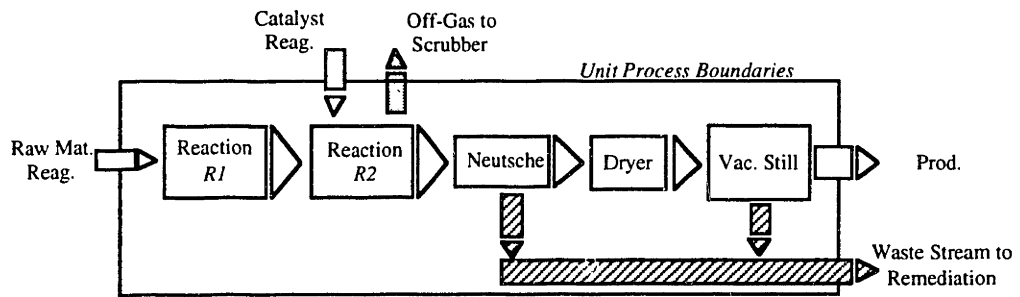


Figure 10.13. Rough Sketch of the Sandmeyer Unit Process.

Estimated unit process capacities and costs are given in Table 10.13.

Table 10.13. Summary of Equipment and Utility Costing Equations for the Sandmeyer Reaction.

Equipment Item:	Utilities	Sizing Equation	Pred. Capacity	Unit Process Eq.	Utilities
<i>Single Product</i>					
reactor(2):	refrig.	$1.3 \times 10^4 (\text{gal}/200)^{0.33}$	90 gal; 136 gal	$2.2 \times 10^4 f^{0.33}$	see below
neutralizer:	—	$0.22 \times 10^4 (\text{gal}/100)^{0.52}$	98.0 gal	$0.22 \times 10^4 f^{0.52}$	—
dryer:	steam	$51.8 \times 10^4 (m^3 + 0.41)$	0.11 m ³	$5.7 \times 10^4 (f + 3.7)$	$1.48 \times 10^4 \bar{f}$
refrigeration eq.:	—	$0.463 \times 10^4 (\text{tons ref})^{0.73}$	4.2 tons ref	$1.3 \times 10^4 f^{0.73}$	—
still:	n.g., steam	$2.28 \times 10^4 (\text{gal}/300)^{0.48}$	14.6 gal	$0.53 \times 10^4 f^{0.48}$	$10.6 \times 10^4 \bar{f}$
collection vess.(2):	—	$0.1 \times 10^4 (\text{gal}/100)^{0.52}$	14.6 gal	$0.09 \times 10^4 f^{0.52}$	—
<i>Additional Costs</i>					
			materials	utilities	misc.
			reagents:	697.0 f	
			labor:		$12.5 f^{0.25}$
			electricity:	$1.4 \bar{f}$	

References and Comments on Table 10.13. **P&T** refers to Peters and Timmerhaus (1980); **P&G** refers to Perry and Green (1984). **Reactors:** glass lined iron kettle; **P&T**, pg. 790 installed cost for January, 1979. **Neutralizer:** mixing tank with agitator; **P&T**, pg. 572 for January, 1979. **Dryer:** rotary vacuum dryer; **P&G**, pg. 20-44 assumed for January, 1979. **Refrigeration Equipment:** **P&G**, pg. 25-76, Marshall and Swift Factor=1000. **Distillation Still:** glass lined vessel with furnace and condenser units (cost factor of 1.2); **P&T**, pg. 790 installed cost for January, 1979. **Collection Vessels:** see Storage Tank. Periodic costs such as utilities, materials, etc. converted to a net present value using a multiplier of 35,000 \$-hr, which corresponds to a project

lifetime of five years. **Labor costs:** operation of multiple small batch units; P&T, pg. 196.

10.9. Supplement: Additional Computational Details

10.9.1. Raw Material, Intermediate, and Product Values

Commodity Values for the compounds in the benzonitrile reaction path network are given in Table 10.14.

Table 10.14. Commodity Values for Compounds in the Case Study

<i>Compound</i>	<i>Sell Price (\$/lb)</i>	<i>Purchase Cost</i>
4-chloroaniline	2.39	2.39
2-nitroaniline	2.01	2.01
2-chloroaniline	1.85	1.85
4-nitroaniline	2.50	2.50
4-chloro-benzonitrile	-5.00	∞
2-nitro-4-chloroaniline	variable	variable
2-chloro-6-nitroaniline	variable	variable
2-chloro-4-nitroaniline	variable	variable
3-nitro-4-chloro-benzonitrile	-5.00	∞
2-nitro-4-chloro-benzonitrile	-5.00	∞
2-nitro-benzonitrile	-5.00	∞
2-chloro-6-nitro-benzonitrile	-5.00	∞
2-chloro-4-nitro-benzonitrile	-5.00	∞
3-amino-4-chloro-benzonitrile	30.33	30.33
2-amino-4-chloro-benzonitrile	32.67	32.67
2-aminobenzonitrile	19.00	19.00
2-amino-6-chloro-benzonitrile	33.03	33.03
2-chloro-4-amino-benzonitrile	26.95	26.95

10.9.2. Waste Stream Costs

Waste stream costs are specific to the positioning of the unit process in the reaction path network. The computed numbers are given in Table 10.15. The index to Table 10.15 is given in Figure 10.14.

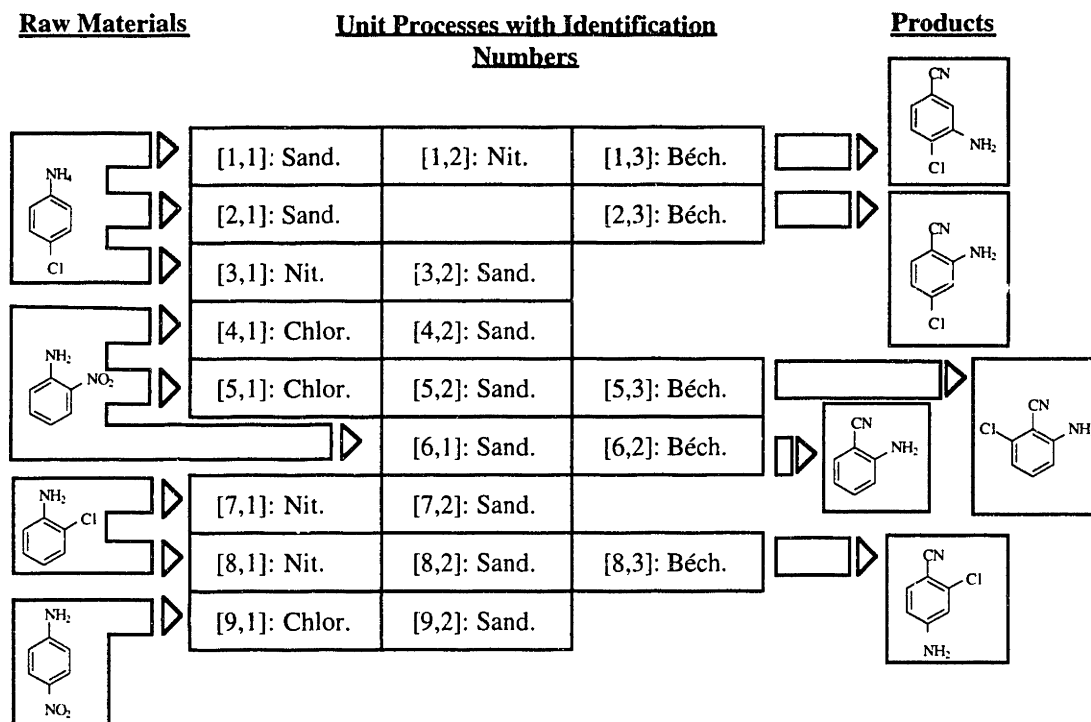


Figure 10.14. Identification Numbers for the Unit Processes for Assignment of Waste Stream Costs.

Table 10.15. Waste Stream Costs (\$/Lb-Mole-Feed).

Unit Process ID	Waste Stream Costs (\$ lb [mol feed] ⁻¹)	
	By-Product Recovered	By-Product Not Recovered
[1,1]	not applic.	37.76
[2,1]	not applic.	37.76
[3,1]	not applic.	61.00
[4,1]	20.00	60.50
[5,1]	20.00	128.00
[6,1]	not applic.	37.76
[7,1]	61.00	102.00
[8,1]	61.00	156.00
[9,1]	not applic.	20.00
[1,2]	61.00	65.00
[3,2]	not applic.	37.76
[4,2]	not applic.	37.76
[5,2]	not applic.	37.76
[6,2]	not applic.	7.10
[7,2]	not applic.	37.76
[8,2]	not applic.	37.76
[9,2]	not applic.	37.76
[1,3]	not applic.	7.10
[2,3]	not applic.	7.10
[5,3]	not applic.	7.10
[8,3]	not applic.	7.10

10.10. Errata

Costs for the rotary dryer are underestimated by a factor of ten as the function $5.7 \times 10^4(f + 3.7)$ has been incorrectly specified as $0.57 \times 10^4(f + 3.7)$ in the mathematical optimizations. The effect on the computational results should be negligible since the dryer is required by each unit process in a network configuration.

Chapter 11

Recommendations for Future Work

This thesis has shown that formal computer-aided design techniques can in fact play a significant role in the systematic identification, evaluation, and screening of alternate chemistries in process development. Integration of relevant problem representations, design algorithms, and chemical information sources in process development creates numerous opportunities to both improve economics and address environmental concerns, to create powerful new algorithms, and to advance the frontiers of chemical information and modeling. Key future directions include the following:

Generalization of evaluation and design objectives to include

- **optimization of process boundaries;**
- **manufacture of multiple products;**
- **decisions with long-term environmental benefits or costs; and**
- **optimal allocation of limited R&D resources.**

The chemical-conversion (chapters two and eight) and unit-process-based tools (chapters seven and ten) developed in this thesis embed such design alternatives and can incorporate a wide range of design objectives. Chapter seven has considered a net present value formulation and an extension to incorporate chance-constrained production targets within the objective function. Analysis of these objectives in chapter ten has shown that a broad range of production strategies can be created for specific industrial requirements. There is a critical need, however, to better account for the items listed above if the best decisions are to be made in process development. Key components of a technical response include

- incorporation of chemical life-cycles into economic measures of process performance;

- treatment of uncertainty in assumptions and parameters when comparing two or more alternate process configurations; and
- development of appropriate problem formulations and solution algorithms capable of handling the objective function forms for large-scale optimization problems.

Exploiting reaction path synthesis as a basis for resource allocation and experimental design that combines the full spectrum of product development activities with the many developmental uncertainties and opportunity costs.

An effective approach must consider the underlying activities that occur in research and development, including model building, experimental measurement, and design calculations. Each of these activities involves the allocation of time, people, and money that can in principle be optimized. Key technical challenges to be addressed include

- the representation of tradeoffs between development of predictive models and direct system measurements in terms of cost, predictive uncertainties (*e.g.*, model variance), and screening capabilities;
- development of computationally efficient resource allocation models that include the large number of investment decisions and the many ways in which they can be configured to form an overall strategy; and
- definition of relevant evaluation criteria for comparing different resource allocation strategies.

Extend the chemical modeling strategy to include mixture properties and more general representations of intermolecular interactions.

Technical approaches for extending the chemical modeling strategy have been discussed in chapters four through six. One of the key challenges that must be addressed is the generalization of interaction potential parameters. The intermolecular potential model developed in chapter six estimates polar attractive and repulsive interactions (based on an analogy to Coulombic point charge interactions) and, to a limited degree, hydrogen-bonding interactions. A number of additional interactions are important in chemical thermodynamics, including charge-transfer and induced dipole interactions, for which the present model is considerably limited. Reaction field methods offer one approach to incorporate the effect of the solvating medium on a molecule. Such approaches are based on embedding the molecule in a dielectric continuum to simulate the effect of surrounding molecules. The charge separation on the molecule gives rise to a corresponding separation

in the dielectric, creating an electric field to which the molecule "reacts" by adjusting its electron density. Development of modern approaches generalizing the original theory due to Onsager are ongoing in several research groups (*e.g.*, Klamt and Schüürmann, 1993). A second approach which is applicable to charge-transfer and hydrogen-bonding interactions is the use of isodesmic reaction sequences. Such reactions have been employed in the prediction of proton affinities and other types of reaction enthalpies that should be useful for the development of correlations predicting hydrogen-bond strengths (chapter four).

Development of property prediction tools as part of a computer-integrated approach to solvent selection.

Screening of alternate reaction-mediating solvent replacements for environmental compliance or improved separability is governed as much by the underlying chemistry as by thermophysical targets. Environmentally benign solvents (*e.g.*, certain classes of esters, sulfolane, *etc.*) tend to involve different functional groups and intermolecular potentials than their EPA-targeted counterparts (*e.g.*, chlorohydrocarbons, aromatics, *etc.*). Consequently, the replacement is rarely a drop-in alternative and frequently changes the underlying reaction process, very possibly at the mechanistic level: reaction rates change, new competing and impurity reactions emerge, and the product distribution (*e.g.*, molecular weight distribution for polymers) is altered. Such variations can be quite important to the properties of performance chemicals such as specialty plastics, surfactants, and many other chemicals. Furthermore, there is essentially zero tolerance for the formation of certain specific impurities (*e.g.*, dioxins). Efforts to address the solvent selection problem in the pharmaceutical and specialty chemical industries are currently underway in several research groups (*e.g.*, Aumond, 1994).

A key modeling problem that can be attacked using the proposed model building strategy (chapters four through six) is the development of linear solvation-energy relationships to predict solvent effects on reaction rates. Consider a bimolecular reaction of the form given in Eq. 1, in which it is assumed that there is a single activation energy barrier.



The choice of solvent can increase or decrease the reaction rate by increasing or decreasing the free energy of solvation of the transition state with respect to the reactant molecules. Such an effect is amenable to transition state theory. Transition state theory allows that the ratio of rates between a solvent and a reference solvent is given by Eq. 2.

$$\log \frac{k}{k_{ref}} = -\frac{\Delta G_{solv}}{RT} \quad (2)$$

where ΔG_{solv} is the difference in the net free energy of solvation given by Eq. 3.

$$\Delta G_{solv} = \Delta G_{A,solv} + \Delta G_{B,solv} - \Delta G_{AB^\ddagger,solv} \quad (3)$$

Equation 3 shows that the key challenge in predicting the solvent effect on the relative reaction rate is to model the thermodynamic interactions between the solvent and the transition state and between the solvent and the reactants. In the case of neutral, apolar reactant molecules proceeding through an isopolar transition state, the free energy due to forming a cavity in the solvent for the reactants and the transition state molecule is often assumed to determine ΔG_{solv} . As a result, it is possible to develop a model for ΔG_{solv} using standard activity coefficient models. Such models have been developed for solutions using the Hildebrand solubility parameter. Furthermore, there is no reason why more sophisticated models such as NRTL or UNIFAC (Reid *et al.*, 1987) could not be applied.

In the general case of dipolar or ionic reactants and transition states, a major problem in developing a model is the wide range of transition state structures, and hence solvent–transition state interactions, that can occur. Group interaction parameters within UNIFAC, for example, probably are not appropriate when the electronic structures in functional groups undergo significant deviations. Furthermore, functional groups with fixed surface areas do not provide a good representation of transition state structures, which often involve "nonstandard" geometric shapes. Due to these quantitative limitations, the classical approach (Reichardt, 1990) to predict solvent–induced effects on reaction rates is fairly limited, dealing mainly with correlations in terms of a functional form of the dielectric constant of the medium.

The approach developed in this research is potentially well-suited to the solvent prediction problem for several reasons. First, the approach could treat the transition state structure explicitly. Specifically, the interaction potential model can handle net charge separation in a transition state molecule. Furthermore, the method can handle arbitrary geometries of intermediate compounds. Second, the constant entropy assumption (or equivalently, Trouton's rule) utilized in deriving the interaction parameters from the normal boiling point is consistent with the linear free-energy treatment, *i.e.*,

$$\begin{aligned} \Delta S_{process} &\propto \Delta H_{process} \text{ or } 0 \Rightarrow \\ \Delta G_{process} &\propto \Delta H_{process} \approx \Delta E_{process} + \text{constant} \end{aligned} \quad (4)$$

An initial result given in chapter six for the Menshutkin reaction demonstrated that parameters derived from pure component properties can be used to predict solvent effects for relatively straightforward reactions such as those governed by the Hugh-Ingold rules (Reichardt, 1990). Building on this initial result, a treatment that directly incorporates interactions between the solvent and the transition state should be broadly applicable, both in developing reaction prediction correlations and in discerning among alternate mechanisms.

Chapter 12

Conclusions

12.1. General Conclusions

The screening and evaluation of process chemistry affect many key decisions in process development and are vital to both economic objectives and environmental concerns. This research has shown that computer-aided design techniques, combining the choice among chemical reaction paths with process design criteria, can improve problem representations, decision-making, and the acquisition of chemical information. Furthermore, such techniques enable a more integrated approach to process development that optimizes the quality of information at each stage of development *via* identification of design tradeoffs, experimental design, and model building. Key research conclusions are as follows:

Construction of the reaction path network for a system of chemistry exposes an extremely rich set of design issues that should be treated in the context of process design, including:

- **optimization of process boundaries;**
- **manufacture of multiple products;**
- **decisions with long-term environmental benefits or costs; and**
- **optimal allocation of limited R&D resources.**

Reaction path networks can, in principle, be developed for many systems of chemistry, including related product lines, total plant chemistry, and even a chemical industry as a whole. The case study for a spectrum of benzonitrile intermediates has demonstrated that a reaction path network facilitates

- identifying process scaleup considerations and high-level design tradeoffs early-on (chapter eight);
- identifying strategic chemicals and technologies for priority testing and development (chapters eight and nine);
- integrating chemical modeling, model building, and direct system measurements to optimize the quality of information in selecting among design alternatives implicit in the reaction path network (chapter nine); and
- formulating decision models based on the hierarchical acquisition of chemical information and refinement of the chemistry and process details (chapters seven and ten).

Molecular calculations using current computing technology and software provide a powerful, practical new avenue for the development of property prediction models in industry.

The model building strategy developed in this research has successfully employed molecular calculations

- as a common source of parameters for a class of thermodynamic property models (chapter six);
- to allow the use of thermodynamic information from readily available pure component properties to estimate additional properties that are difficult and expensive to measure (chapter six);
- to reduce the number of parameters in comparison with group contribution methods (chapter six);
- to allow estimation of property differences between isomeric compounds that are completely averaged-out by group contribution methods (nine); and
- to exploit state-of-the-art computers and computational techniques that are now becoming more widely available, both in universities and in industry.

As demonstrated by the formal mathematical programming strategy (chapters seven and ten), the most economic or environmentally favorable design can involve multiple products and highly integrated configurations of reaction paths and processes.

Two key implications are as follows:

- Production decisions, management practices, or regulations that focus only on single products or narrowly defined objectives can miss critical synergies or tradeoffs in design.
- Process flexibility must be more broadly defined to include not only the range of operation of a process, but also the raw materials, intermediates, and chemical technology which can respond to changes in market demand, raw material costs, and environmental regulations.

12.2. Technical Conclusions

A basic goal was to systematically generate design alternatives arising from fundamental changes in raw materials, intermediates, and chemical conversions. An implicit technical goal was to identify the crucial differentiating criteria to isolate the most promising reaction paths from random permutations of chemical conversion sequences. To address these goals, a formal reaction path synthesis algorithm combining retrosynthetic analysis and reaction prediction was created. Several classes of industrial aromatic chemistry, including a large-scale case study for a set of five benzonitrile dye intermediates and six industrial conversions, served as testbeds for the algorithm. Combinations of reaction rules with bounds derived from linear free-energy models were shown to provide a strong filter, even in the presence of significant model deviations. In the case study, for example, the total number of precursors was reduced from 687 to 51. The precursors combined to form both existing industrial routes and novel integrated routes. Among the new routes, nitrations and chlorinations of substituted anilines were found to yield mixtures of reaction products that could be separated and sent to multiple production lines. Several of the routes had highly integrated chemistries and would have been difficult to find without a systematic computational approach. The consideration of multiple products in evaluating the proposed routes allowed several production strategies to be identified, including (i) manufacture of multiple products from a common base of chemistry; (ii) choice among process boundaries, including in-sourcing of several intermediates at an economic scale of production; and (iii) choices between separation costs and waste disposal costs.

A realistic evaluation of reaction path alternatives must quantify the common resource and technology requirements as well as synergies between product lines. A unit-process-based decomposition was developed in which process tasks, material balances, and energy balances were aggregated around chemical conversions to form a flowsheet.

The decomposition allowed formulation of an efficient mixed-integer linear program (derived as an approximation to a separably nonconvex mixed-integer nonlinear program). A chance-constrained augmentation of the formulation was then developed to incorporate uncertainty in production targets directly within the objective function. The case study mentioned above, subjected to a net present value design objective, required approximately one hundred binary and five hundred continuous variables (initial linear programming relaxations were within 2% of the global optimum). Whereas the traditional process configurations were obtained in certain limits, new process configurations were also synthesized over a wide range of realistic intermediate prices and waste stream costs. A common feature among the new configurations was the separation of usable reaction by-products with any excess sold as intermediate products. Thus, process flexibility did not suffer but was actually enhanced through the choice of multipurpose raw materials and chemical conversions. In this regard, extension of the case study to the chance-constrained formulation yielded more promising configurations than those obtained from an expected value criterion (the expected value criterion tended to minimize the volume of products produced).

A key challenge in each of the scoping and design activities discussed above is the acquisition of chemical information at an appropriate level of accuracy and cost. To address limitations in group contribution approaches to model building, a strategy was created that exploits computed molecular surface areas and electronic density measures within an athermal lattice potential. Such an approach was shown to be sensitive to structural differences that are traditionally averaged out by group contribution methods. A data set of approximately 160 compounds was constructed for polar, hydrogen-bonding, and charge-transferring compounds and included a molecular weight distribution of approximately 50–200, boiling point range of 80–500°C, and a molecular surface area range of 120–300Å². Coefficients of determination for the normal boiling points ranged 0.97–0.99, with lower accuracy attributable to orientation-dependent hydrogen-bonding and charge-transfer interactions. Accurate correlation of the normal boiling point (about one-third smaller variance and 25% fewer parameters compared with Joback's method) allowed intrinsic interaction energies to be extracted for development of log-linear models of partition coefficients and linear solvation-energy relationships. A property prediction module incorporating the new models was applied to the case study discussed above. Whereas current group contribution methods failed to distinguish the boiling points between the isomeric reaction by-products, the new model predicted relative volatilities *ca.* 2.0. The property package was integrated into a batch distillation design module and design estimates performed.

12.3. Concluding Summary

In summary, the original premise of this research was that a successful response to economic objectives and the environment must combine the choice of system chemistry with process design in a process development framework. This thesis has shown that formal computer-aided design techniques can in fact play a significant role in the systematic identification, evaluation, and screening of alternate chemistries in process development. Integration of relevant problem representations, design algorithms, and chemical information sources in process development opens up a broad range of industrial opportunities to improve both economics and the environment, to create powerful new algorithms, and to advance the frontiers of chemical information and modeling.

Appendix 1: Compound Databank

Table A1.1. Compound Databank

No.	Compound	Filename	Pure Component Properties					Interaction Potential Descriptors				
			T_c	V_c	P_c	V_{mol}	$T_{nbp}^{pred.}$	$T_{nbp}^{meas.}$	dispers.	polar attr.	polar repul.	hyd. bond
1	C ₆ H ₆	mono.01	562.1	259.0	48.9	88.2	369.2	353.2	159.8	-3.52	11.04	0.000
2	C ₆ H ₅ NO ₂	mono.02	-	-	-	-	488.0	484.0	192.8	-3.64	7.49	0.000
3	C ₆ H ₅ NH ₂	mono.03	699.0	274.0	53.1	91.1	465.0	457.5	169.2	-5.84	12.94	0.000
4	C ₆ H ₅ OH	mono.04	694.2	229.0	61.3	88.9	455.2	455.0	163.2	-4.13	8.57	-0.393
5	C ₆ H ₅ CN	mono.05	699.4	310.0	42.2	102.1	464.0	464.0	194.9	-3.13	7.70	0.000
6	C ₆ H ₅ Cl	mono.06	632.4	308.0	45.2	101.8	408.3	404.9	181.7	-2.19	6.99	0.000
7	C ₆ H ₅ CH ₃	mono.07	591.8	316.0	41.0	106.2	388.5	383.8	183.7	-2.89	10.55	0.000
8	C ₆ H ₅ F	mono.08	560.1	269.0	45.5	93.8	-	357.9	-	-	-	-
9	1,2-C ₆ H ₄ (CH ₃) ₂	diout1.09	630.3	369.0	37.3	120.7	403.6	417.6	201.8	-2.39	10.08	0.000
10	1,3-C ₆ H ₄ (CH ₃) ₂	diout1.10	617.1	376.0	35.4	122.9	407.6	412.3	207.0	-2.34	10.23	0.000
11	1,4-C ₆ H ₄ (CH ₃) ₂	diout1.11	616.2	379.0	35.1	123.3	407.8	411.5	208.0	-2.33	10.30	0.000
12	1,2-C ₆ H ₄ CH ₃ NH ₂	diout1.12	694.0	343.0	37.5	107.4	454.6	473.5	190.3	-4.60	12.21	0.000
13	1,3-C ₆ H ₄ CH ₃ NH ₂	diout1.13	709.0	343.0	41.5	108.4	458.7	467.6	193.0	-4.30	11.34	0.000
14	1,4-C ₆ H ₄ CH ₃ NH ₂	diout1.14	667.0	353.0	43.8	111.2	460.9	473.7	193.9	-4.24	11.12	0.000
15	1,2-C ₆ H ₄ Cl ₂	diout2.15	697.3	360.0	41.0	112.6	443.5	452.0	200.5	-1.32	4.20	0.000
16	1,3-C ₆ H ₄ Cl ₂	diout2.16	684.0	359.0	38.0	114.1	449.0	446.0	202.9	-1.26	3.93	0.000
17	1,4-C ₆ H ₄ Cl ₂	diout2.17	685.0	372.0	39.0	117.8	447.8	447.3	203.1	-1.25	3.99	0.000
18	1,2-C ₆ H ₄ CH ₃ OH	diout1b.18	697.6	282.0	50.1	105.2	469.1	464.2	187.4	-3.49	8.25	-0.296
19	1,3-C ₆ H ₄ CH ₃ OH	diout1b.19	705.8	309.0	45.6	104.5	469.2	475.4	186.4	-3.50	8.26	-0.348

Table A1.1. Compound Databank, continued

No.	Compound	Filename	Pure Component Properties					Interaction Potential Descriptors				
			T_c	V_c	P_c	V_{mol}	$T_{nbp}^{pred.}$	$T_{nbp}^{meas.}$	dispers.	polar attr.	polar repul.	hyd. bond
20	1,4-C ₆ H ₄ CH ₃ OH	diout.1b.20	704.6	300.0	51.5	106.1	468.9	475.1	187.2	-3.45	8.20	-0.342
21	MeCH ₂ Me	alkanout.21	369.8	203.0	42.5	75.8	253.3	231.0	134.2	0.00	5.05	0.000
22	Me(CH ₂) ₂ Me	alkanout.22	425.2	255.0	38.0	100.3	280.3	272.7	159.8	0.00	6.15	0.000
23	Me ₃ CH	alkanout.23	408.2	263.0	36.5	104.3	276.5	261.4	155.3	0.00	5.89	0.000
24	C(Me) ₄	alkanout.24	433.8	303.0	32.0	122.2	298.8	282.6	173.5	0.00	6.47	0.000
25	Me(CH ₂) ₃ Me	alkanout.25	469.7	304.0	33.7	115.3	307.2	309.2	185.8	0.00	7.28	0.000
26	Me ₂ CHCH ₂ Me	alkanout.26	460.4	306.0	33.9	116.5	299.8	301.0	177.5	0.00	6.84	0.000
27	Me(CH ₂) ₄ Me	alkanout.27	507.5	370.0	30.1	130.8	332.8	341.9	211.0	0.00	8.42	0.000
28	Me(CH ₂) ₅ Me	alkanout.28	540.3	432.0	27.4	146.5	359.3	371.6	236.4	0.00	9.51	0.000
29	Me(CH ₂) ₆ Me	alkanout.29	568.8	492.0	24.9	162.4	386.0	398.8	261.8	0.00	10.60	0.000
30	CCl ₃ H	alkclout.31	536.4	238.9	53.7	80.2	360.3	334.3	150.8	-0.03	0.17	0.000
31	Cl-CH ₂ -Me	alkclout.32	460.4	199.0	52.7	72.0	289.8	285.4	131.5	-0.14	2.87	0.000
32	Me(CH ₂) ₂ CH ₂ -Cl	alkclout.33	542.0	312.0	36.8	104.5	334.0	351.6	177.2	-0.15	5.09	0.000
33	MeCH ₂ CHClMe	alkclout.34	520.6	305.0	39.5	106.1	341.8	341.4	180.3	-0.15	4.94	0.000
34	Me ₃ CCl	alkclout.35	507.0	295.0	39.5	110.0	329.8	324.0	173.0	-0.15	4.90	0.000
35	Cl ₂ CHMe	alkclout.36	523.0	236.0	50.7	84.8	343.1	330.5	152.2	-0.13	1.70	0.000
36	ClCH ₂ CH ₂ Cl	alkclout.37	566.0	225.0	53.7	79.2	348.9	356.7	154.8	-0.17	1.75	0.000
37	Cl ₂ CHCH ₂ Cl	alkclout.38	606.0	294.0	51.4	92.6	385.8	386.7	171.4	-0.09	0.99	0.000
38	MeCH ₂ NH ₂	aminout.41	456.4	182.0	56.4	66.0	300.6	289.7	120.1	-2.24	7.23	0.000
39	MeNHMe	aminout.42	437.7	187.0	53.1	68.7	287.7	280.0	124.3	-0.95	4.62	0.000
40	MeCH ₂ CH ₂ NH ₂	aminout.43	497.0	233.0	48.1	82.4	323.7	321.7	145.3	-2.18	8.34	0.000

Table A1.1. Compound Databank, continued

No.	Compound	Filename	Pure Component Properties					Interaction Potential Descriptors				
			T_c	V_c	P_c	V_{mol}	$T_{nbp}^{pred.}$	$T_{nt.}^{meas.}$	dispers.	polar attr.	polar repul.	hyd. bond
41	Me ₂ CHNH ₂	aminout.44	478.1	221.0	45.4	85.9	303.6	305.6	144.3	-1.36	7.04	0.000
42	Me ₃ N	aminout.45	433.3	254.0	40.9	93.4	289.4	276.0	152.8	0.00	4.79	0.000
43	NH ₂ CH ₂ CH ₂ NH ₂	aminout.46	593.0	206.0	62.8	67.1	398.7	390.4	131.5	-5.13	10.92	0.555
44	Me(CH ₂) ₃ NH ₂	aminout.47	531.9	288.0	42.0	98.9	349.6	349.5	170.8	-2.16	9.41	0.000
45	MeCH ₂ NHCH ₂ Me	aminout.48	496.5	301.0	37.1	103.4	335.5	328.6	167.7	-0.93	6.28	0.000
46	MeCN	cyanout.51	545.5	173.0	48.3	52.6	355.5	354.8	117.7	-1.94	2.58	0.000
47	MeCH ₂ CN	cyanout.52	564.0	229.0	41.8	70.5	374.2	370.5	144.3	-1.71	3.61	0.000
48	Me(CH ₂) ₂ CN	cyanout.53	582.2	285.0	37.9	87.2	406.5	391.1	170.0	-1.77	4.56	0.000
49	1,2-C ₆ H ₄ (NF ₂) ₂	diaminout.54	-	-	-	-	518.1	525.0	177.4	-7.42	15.22	0.000
50	1,3-C ₆ H ₄ (NH ₂) ₂	diaminout.55	-	-	-	-	573.1	560.0	178.2	-8.70	15.67	0.000
51	1,4-C ₆ H ₄ (NH ₂) ₂	diaminout.56	-	-	-	-	543.3	540.0	178.6	-7.86	15.07	0.000
52	CH ₃ NO ₂	nitrou.57	588.0	63.1	173.2	-	-	374.3	120.3	-1.23	2.23	0.000
53	1,2-C ₆ H ₄ NH ₂ NO ₂	diout1.60	-	-	-	-	493.8	495.6	210.0	-3.22	7.74	0.000
54	1,3-C ₆ H ₄ NH ₂ NO ₂	diout1.61	-	-	-	-	511.0	505.0	217.5	-3.16	7.30	0.000
55	1,4-C ₆ H ₄ NH ₂ NO ₂	diout1b.62	-	-	-	-	509.1	511.7	216.9	-3.20	7.47	0.000
56	1,2-C ₆ H ₄ CNCl	diout2.63	-	-	-	-	486.0	505.2	213.5	-1.91	4.70	0.000
57	1,2-C ₆ H ₄ NH ₂ Cl	diout2.64	-	-	-	-	472.6	482.0	189.3	-3.78	8.49	0.000
58	1,3-C ₆ H ₄ NH ₂ Cl	diout2.65	-	-	-	-	489.7	501.7	190.7	-4.10	8.53	0.000
59	1,2-C ₆ H ₄ NO ₂ Cl	diout2.66	-	-	-	-	508.2	519.0	213.3	-2.34	4.56	0.000
60	1,3-C ₆ H ₄ NO ₂ Cl	diout2.67	-	-	-	-	512.1	508.8	214.8	-2.24	4.17	0.000

Table A1.1. Compound Databank, continued

No.	Compound	Filename	Pure Component Properties					Interaction Potential Descriptors				
			T_c	V_c	P_c	V_{mol}	$T_{nbp}^{pred.}$	$T_{nbp}^{meas.}$	dispers.	polar attr.	polar repul.	hyd. bond
61	1,4-C ₆ H ₄ NO ₂ Cl	diout.68	-	-	-	-	511.5	515.2	214.3	-2.28	4.28	0.000
62	1,3-C ₆ H ₄ NH ₂ CN	diout.69	-	-	-	-	546.1	563.8	204.2	-5.06	9.31	0.000
63	HCOOMe	gout.71	487.2	172.0	60.0	61.7	312.3	304.9	125.4	-0.84	2.87	0.000
64	HCOOCH ₂ Me	gout.72	508.5	229.0	47.4	79.9	346.3	327.4	155.1	-0.92	4.18	0.000
65	MeCOOMe	gout.73	506.8	228.2	46.9	79.3	331.3	330.4	145.8	-0.89	4.04	0.000
66	MeCOOCHCH ₂	gout.74	525.0	265.0	43.5	92.4	352.7	346.0	168.1	-1.35	6.50	0.000
67	MeOOCCHCH ₂	gout.75	536.0	265.0	42.0	90.1	347.5	353.5	166.3	-1.35	6.63	0.000
68	MeCOOCH ₂ Me	gout.76	523.2	286.0	38.3	97.8	361.7	350.3	175.2	-0.93	5.43	0.000
69	MeCH ₂ COOMe	gout.77	530.6	282.0	40.0	96.3	346.0	352.8	175.0	-0.88	6.29	0.000
70	HCOO(CH ₂) ₂ Me	gout.78	538.0	285.0	40.6	96.7	374.7	354.1	180.7	-0.95	5.29	0.000
71	CH ₂ CHCOOCH ₂ Me	gout.79	552.0	320.0	37.4	108.7	369.3	373.0	194.5	-1.36	8.36	0.000
72	MeCOOCOMe	gout.80	569.0	290.0	46.8	93.9	403.1	413.2	179.7	-1.54	5.12	0.000
73	HCOOCH ₂ CHMe ₂	gout.81	555.0	352.0	37.3	115.4	375.0	371.4	199.0	-1.01	7.47	0.000
74	MeCOO(CH ₂) ₂ Me	gout.82	549.4	345.0	33.3	115.1	390.1	374.7	201.4	-0.95	6.58	0.000
75	MeCH ₂ COOCH ₂ Me	gout.83	546.0	345.0	33.6	114.1	368.9	372.0	203.8	-0.90	8.02	0.000
76	Me(CH ₂) ₂ COOMe	gout.84	554.4	340.0	34.8	113.7	375.9	375.9	201.3	-0.89	7.30	0.000
77	Me ₂ CHCOOMe	gout.85	540.8	339.0	34.3	114.6	371.3	365.4	198.6	-0.87	7.21	0.000
78	MeCOO(CH ₂) ₃ Me	gout.86	579.0	400.0	31.4	129.4	419.4	399.3	226.8	-0.96	7.53	0.000
79	MeCOOCH ₂ CHMe ₂	gout.87	564.0	414.0	30.2	132.8	404.2	389.7	226.1	-0.98	8.46	0.000
80	Me(CH ₂) ₂ COOCH ₂ Me	gout.88	569.0	421.0	29.6	132.2	393.5	394.7	227.8	-0.86	8.98	0.000

Table A1.1. Compound Databank, continued

No.	Compound	Filename	Pure Component Properties					Interaction Potential Descriptors				
			T_c	V_c	P_c	V_{mol}	$T_{nbp}^{pred.}$	$T_{nbp}^{meas.}$	dispers.	polar attr.	polar repul.	hyd. bond
81	Me ₂ CHCOOCH ₂ Me	gout.89	555.0	421.0	29.7	133.7	385.2	383.2	224.3	-0.81	8.98	0.000
82	MeOCCOOMe	gout.90	628.0	307.0	39.8	102.7	428.6	436.5	191.9	-1.25	3.99	0.000
83	C ₆ H ₅ COOMe	gout.91	692.0	396.0	36.4	125.4	480.3	472.2	223.1	-3.44	10.70	0.000
84	MeCOOH	gout.92	592.7	171.0	57.9	57.3	370.6	391.1	117.8	-1.72	2.41	-0.722
85	CH ₂ CHCOOH	gout.93	615.0	210.0	56.7	68.6	401.6	414.0	134.3	-2.28	3.74	-0.618
86	MeCH ₂ COOH	gout.94	612.0	222.0	54.0	74.6	400.3	414.5	142.9	-1.84	3.43	-0.619
87	Me(CH ₂) ₂ COOH	gout.95	628.0	290.0	52.7	92.0	432.3	437.2	168.6	-1.94	4.33	-0.535
88	Me ₂ CHCOOH	gout.96	609.0	292.0	40.5	91.0	427.0	427.9	166.3	-1.91	4.26	-0.500
89	Me(CH ₂) ₃ COOH	gout.97	651.0	340.0	38.0	108.7	462.9	459.5	194.6	-2.00	5.28	-0.456
90	C ₆ H ₅ COOH	gout.98	752.0	341.0	45.6	113.6	535.0	523.0	191.7	-4.50	7.68	-0.364
91	MeCH ₂ OH	gout.101	513.9	167.1	61.4	58.4	330.9	351.4	115.0	-1.11	2.88	-0.758
92	Me(CH ₂) ₂ OH	gout.102	536.8	219.0	51.7	74.8	356.1	370.3	140.4	-1.20	4.00	-0.606
93	Me ₂ CHOH	gout.103	508.3	220.0	47.6	76.5	351.1	355.4	140.7	-1.20	4.27	-0.562
94	(HOCH ₂) ₂	gout.104	645.0	186.0	77.0	55.7	455.4	470.5	121.5	-1.82	2.29	-2.987
95	MeCHOH(CH ₂ OH)	gout.105	625.0	237.0	60.7	73.5	463.4	460.5	146.0	-2.10	3.62	-2.155
96	(HOCH ₂) ₂ CH ₂	gout.106	658.0	241.0	59.5	72.3	465.7	487.6	147.0	-2.02	3.29	-2.134
97	Me(CH ₂) ₃ OH	gout.107	563.1	275.0	44.2	91.5	384.2	390.9	166.2	-1.24	5.00	-0.520
98	MeCH ₂ CH(OH)Me	gout.108	536.1	269.0	41.8	91.8	366.1	372.7	162.0	-1.18	5.37	-0.448
99	Me ₂ CHCH ₂ OH	gout.109	547.8	273.0	43.0	92.4	379.4	381.0	162.4	-1.22	4.84	-0.517
100	Me ₃ COH	gout.110	506.2	275.0	39.7	94.2	356.4	355.5	160.8	-1.09	5.47	-0.387

Table A1.1. Compound Databank, continued

No.	Compound	Filename	Pure Component Properties					Interaction Potential Descriptors				
			T_c	V_c	P_c	V_{mol}	$T_{nbp}^{pred.}$	$T_{nbp}^{meas.}$	dispers.	polar attr.	polar repul.	hyd. bond
101	Me(CH ₂) ₄ OH	gout.111	588.2	326.0	39.1	108.2	410.6	411.1	191.6	-1.27	6.07	-0.450
102	MeCH ₂ CHMeCH ₂ OH	gout.112	571.0	322.0	33.4	107.7	395.2	401.9	181.0	-1.24	5.71	-0.420
103	MeCHMe(CH ₂) ₂ OH	gout.113	579.4	329.0	38.0	108.9	399.6	405.2	181.4	-1.26	5.65	-0.469
104	MeCH ₂ C(MeOH)Me	gout.114	545.0	319.0	39.5	109.0	374.5	375.5	181.5	-1.08	6.36	-0.296
105	(HOCH ₂) ₂ CHOH	gout.116	726.0	255.0	66.8	73.0	583.1	563.0	152.2	-2.65	3.16	-4.555
106	Me(CH ₂) ₅ OH	gout.117	611.0	381.0	40.5	124.8	438.0	430.2	217.4	-1.30	7.15	-0.398
107	(HO(CH ₂) ₂) ₂ O	gout.118	681.0	316.0	47.0	95.1	537.2	519.0	184.4	-2.33	4.46	-2.469
108	C ₆ H ₅ CH ₂ OH	gout.119	720.2	334.0	44.0	103.8	489.6	478.6	189.5	-4.07	9.04	-0.369
109	Me(CH ₂) ₆ OH	gout.120	633.0	435.0	30.4	141.4	464.1	449.8	242.7	-1.32	8.22	-0.357
110	Me(CH ₂) ₇ OH	gout.121	652.5	490.0	28.6	157.6	491.0	468.3	268.5	-1.34	9.33	-0.321
111	Me(CH ₂) ₅ CH(OH)Me	gout.122	637.0	494.0	27.0	158.6	469.2	452.0	263.5	-1.20	9.58	-0.236
112	HO(CH ₂) ₂ NH ₂	gout.123	614.0	196.0	44.5	60.1	441.3	443.5	126.3	-3.40	5.89	-1.715
113	CHOMe	aldeout.131	461.0	154.0	55.7	56.7	285.0	294.0	113.8	-0.75	3.04	0.000
114	CHOCH ₂ Me	aldeout.132	515.0	223.0	63.3	72.9	316.5	321.0	140.7	-0.78	4.09	0.000
115	Me ₂ CO	aldeout.133	508.1	209.0	47.0	73.5	303.3	329.0	138.3	-0.80	4.73	0.000
116	MeCOCH ₂ Me	aldeout.134	536.8	267.0	42.1	89.6	328.1	352.7	159.5	-0.81	5.50	0.000
117	HCO(CH ₂) ₂ Me	aldeout.135	545.4	278.0	53.8	89.9	345.9	348.0	166.2	-0.80	5.06	0.000
118	HCOCH(CH ₃) ₂	aldeout.136	513.0	274.0	41.5	91.4	338.5	337.0	162.4	-0.74	4.96	0.000
119	HCO(CH ₂) ₃ Me	aldeout.137	554.0	333.0	35.4	106.3	374.0	376.0	191.3	-0.82	6.10	0.000
120	ClCOMe	aldeout.139	508.0	204.0	58.7	71.1	335.3	323.9	136.9	-0.48	1.58	0.000

Table A1.1. Compound Databank, continued

No.	Compound	Filename	Pure Component Properties					Interaction Potential Descriptors						
			T_c	V_c	P_c	V_{mol}	$T_{nbp}^{pred.}$	$T_{nbp}^{meas.}$	dispers.	polar attr.	polar repul.	hyd. bond		
141	1,3,5-C ₆ H ₃ (NO ₂) ₃	raout.174	-	-	-	-	-	-	-	-	-	-	-	-
142	1,2,4,6-C ₆ H ₂ CH ₃ (NO ₂) ₃	raout.175	-	-	-	-	-	-	-	-	-	-	-	-
143	1,4-C ₆ H ₄ CINH ₂	raout.181	-	-	-	-	484.8	-	191.1	-3.96	8.48	0.000	0.000	0.000
144	1,2-C ₆ H ₄ NH ₂ NO ₂	raout.182	-	-	-	-	553.3	-	199.3	-5.18	8.68	0.000	0.000	0.000
145	1,4-C ₆ H ₄ NH ₂ NO ₂	raout.183	-	-	-	-	572.4	-	201.8	-5.55	8.83	0.000	0.000	0.000
146	1,4-C ₆ H ₄ ClCN	raout.184	-	-	-	-	497.6	-	216.6	-1.98	4.49	0.000	0.000	0.000
147	1,2,4-C ₆ H ₃ NH ₂ NO ₂ Cl	raout.185	-	-	-	-	562.0	-	221.6	-3.35	5.08	0.000	0.000	0.000
148	1,2,6-C ₆ H ₃ NH ₂ NO ₂ Cl	raout.186	-	-	-	-	549.8	-	218.7	-3.25	5.23	0.000	0.000	0.000
149	1,2,4-C ₆ H ₃ NH ₂ ClNO ₂	raout.187	-	-	-	-	567.6	-	222.0	-3.47	5.13	0.000	0.000	0.000
150	1,3,4-C ₆ H ₃ CNNO ₂ Cl	raout.188	-	-	-	-	579.3	-	248.8	-1.87	2.50	0.000	0.000	0.000
151	1,2,4-C ₆ H ₃ CNNO ₂ Cl	raout.189	-	-	-	-	566.2	-	243.2	-1.75	2.39	0.000	0.000	0.000
152	1,2-C ₆ H ₄ CNNO ₂	raout.190	-	-	-	-	550.8	-	222.2	-3.00	4.82	0.000	0.000	0.000
153	1,2,6-C ₆ H ₃ CNNO ₂ Cl	raout.191	-	-	-	-	558.5	-	239.5	-1.81	2.63	0.000	0.000	0.000
154	1,2,4-C ₆ H ₃ CNCINO ₂	raout.192	-	-	-	-	573.5	-	245.7	-1.84	2.45	0.000	0.000	0.000
155	1,3,4-C ₆ H ₃ CNNH ₂ Cl	raout.193	-	-	-	-	553.2	-	224.1	-3.29	5.73	0.000	0.000	0.000
156	1,2,4-C ₆ H ₃ CNNH ₂ Cl	raout.194	-	-	-	-	544.9	-	223.7	-3.19	5.89	0.000	0.000	0.000
157	1,2-C ₆ H ₄ CNNH ₂	raout.195	-	-	-	-	518.8	-	202.3	-4.60	9.47	0.000	0.000	0.000
158	1,2,6-C ₆ H ₃ CNNH ₂ Cl	raout.196	-	-	-	-	533.5	-	220.4	-3.15	6.13	0.000	0.000	0.000
159	1,2,4-C ₆ H ₃ CNCINH ₂	raout.197	-	-	-	-	544.5	-	222.5	-3.27	6.05	0.000	0.000	0.000

Appendix 2: Polaroid Case Study Report

POLAROID PROJECT REPORT #2:

**Hydroxy methylation–Dehydrogenation Synthesis of
DMB; and Summary of Available Literature on
Synthesis of DMB *via* Vilsmeier–Haack
Formylation**

March 3, 1994

Jonathan P. Knight
Chemical Engineering Dept., MIT
(617) 253–0285

Submitted to:

Neil Kaufman
Polaroid Corporation

Summary

This report focuses on alternatives to the Vilsmeier–Haack formylation. Given below are several possible synthesis routes and some of the computational analysis that were used to eliminate certain candidate routes from consideration. The most promising route appears to be titanium alcoholate-catalyzed dehydrogenation of 2,5-dimethoxybenzenemethanol to DMB (**Di**Methoxy**B**enzaldehyde) due to its relatively inexpensive raw materials, compatibility with nonchlorinated solvents, and use of a nontoxic catalyst. Several other routes are also provided for consideration.

Literature on the synthesis of DMB by Vilsmeier–Haack formylation are listed. Low yields of DMB are reported, suggesting the need for optimizations of the solvent and reagent. Finally, patent citations are provided for several DMB processes.

Hydroxymethylation–Dehydrogenation Procedure

The two-step route shown in Figure A2.1, a formaldehyde condensation followed by dehydrogenation to the benzaldehyde, represents the strongest candidate reaction path to DMB identified thus far.

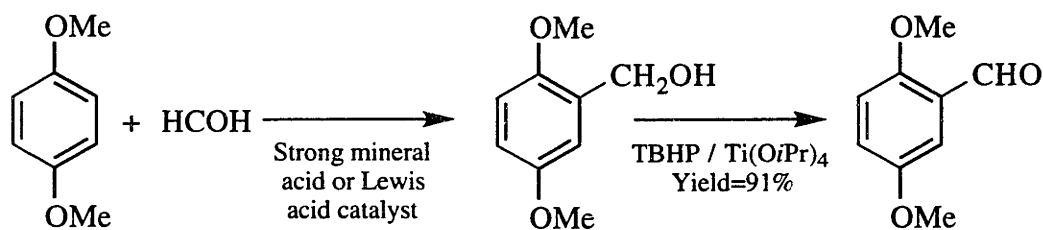


Figure A2.1. Synthesis of DMB Based on Dehydrogenation of Dimethoxy Benzyl Alcohol

The second step is a titanium-catalyzed selective dehydrogenation using *tert*-butyl hydroperoxide (TBHP). The yield of 91% is for the compound shown (Krohn *et al.*, 1990). The reaction can be carried out in one of several solvents; dichloromethane, acetone, tetrahydrofuran, and ethyl acetate have been investigated. The amount of titanium catalyst is catalytic—about 10 mole%. Combining this fact with relatively low cost and safe handling, the titanium alcoholate catalyst is considered generally superior to alternate catalysts such as MnO₂ and CrO₂. The above-mentioned paper provides a detailed account of the synthesis procedure. Relevant pages from the article are included in the appendix.

The formaldehyde condensation is proposed on the basis of what is known about chloromethylation procedures and the Lederer–Manasse reaction (hydroxyalkylation of phenols). In the chloromethylation of dimethoxybenzene, yields as high as 95% have been

reported (Olah and Kuhn, 1964) using the synthesis procedure shown approximately in Figure A2.2.

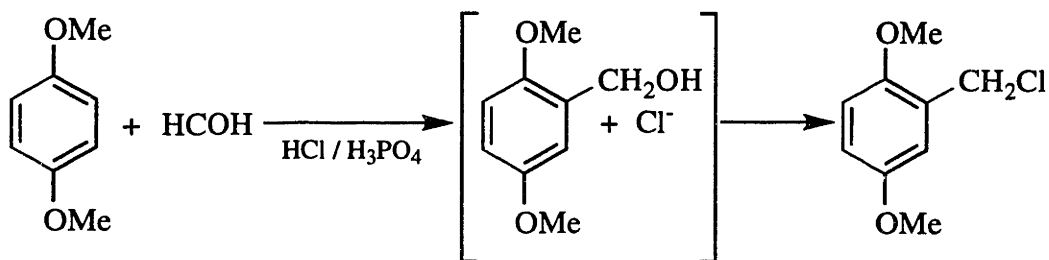


Figure A2.2. Chloromethylation Procedure Suggesting the Possibility of A High-Yield Approach to Hydroxymethylation.

Through the use of either a strong mineral acid (*e.g.*, H_2SO_4 , H_3PO_4) or a Lewis acid catalyst (*e.g.*, ZnCl_2) it may be possible to use formaldehyde, an inexpensive raw material, to generate the substituted benzyl alcohol. If a Lewis acid is required, additional recovery procedures may be involved. The reaction is expected to require careful control to avoid formation of Bakelite-type polymers.

Chloromethylene Dibenzoate Condensation and Related Reactions

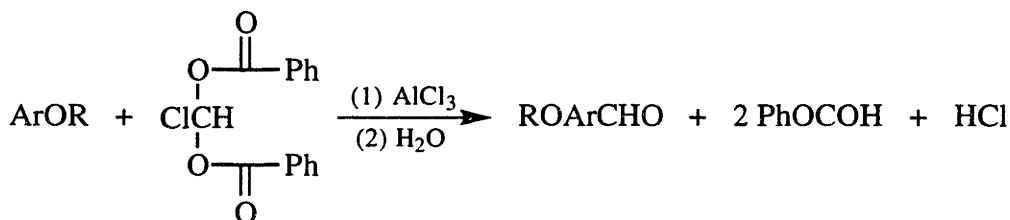


Figure A2.3. Chloromethylene Dibenzoate Condensation.

The condensation procedure has been illustrated in the formylation of anisaldehyde in benzene solvent (Ferguson, 1946). Reported yields are better than 80%.

A related reaction that may exhibit more favorable economics is the condensation of triethyl orthoformate, $\text{CH}(\text{OEt})_3$, in the presence of AlCl_3 (Buehler and Pearson, 1977).

Formylation by Formyl Fluoride

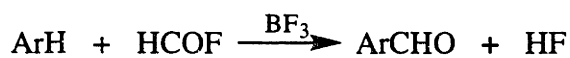


Figure A2.4. Formylation by Formyl Fluoride.

The procedure is not pursued here due to the high cost of catalyst (*ca.* 6-\$/LB) relative to other Lewis acid catalysts and special handling considerations associated with HF.

Fragmentation of Secondary Alcohols with Lead Tetraacetate

A multi-step synthesis of DMB has been illustrated (Shankaran and Rao, 1980) that involves relatively straightforward steps (Figure A2.5). The key step is the last one, which proceeds by a free-radical mechanism evolving benzyl acetate as a by-product.

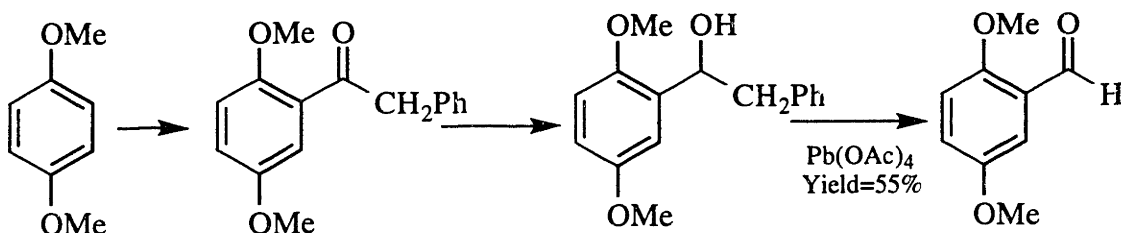


Figure A2.5. Fragmentation of a Secondary Alcohol.

The route is not expected to exhibit favorable economics due to yield losses and raw material costs. The yield of 55% is for the compound shown.

Manganese(III)-Mediated Formylation in Presence of Malonic Acid

Manganese(III) acetate, $[\text{Mn}_3\text{O}(\text{AcO})_6(\text{OAc})(\text{HOAc})] \cdot 5\text{H}_2\text{O}$, under suitable conditions, will undergo a free radical reaction with malonic acid, $\text{CH}_2(\text{CO}_2\text{H})_2$, to yield dicarboxymethyl radicals. These radicals initiate a relatively complex sequence of events that, in the case of highly activated aromatic compounds, leads to a one-step formylation of the aromatic nucleus (Figure A2.6). In less activated substrates, the reaction proceeds in a reduced capacity that tends to produce carboxylic acid substitutions.

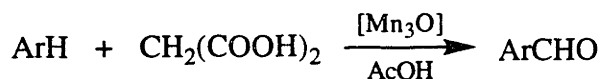


Figure A2.6. Manganese Mediated Formylation

In a systematic study of naphthalenes, anthracenes, pyrenes, and methoxybenzenes (Nishino *et al.*, 1989), an upper bound of 7.8 eV on the ionization potential of aromatic substrates has been established, above which the reaction depicted in Figure A2.6 is not expected to proceed to any useful extent under the reaction conditions studied. Since the reaction has not previously been carried out on 1,4-dimethoxybenzene, it is desirable to evaluate the ionization potential of this molecule. Open shell *ab initio* molecular orbital calculations (*i.e.*, unrestricted Hartree-Fock calculations) have been performed at the

3-21G* level, with a second order Møller–Plesset perturbation, on anisole, 1,4-dimethoxybenzene, and their ionized radicals. The positively charged radicals are assumed to be ground state doublets. The results are summarized in Figure A2.7.

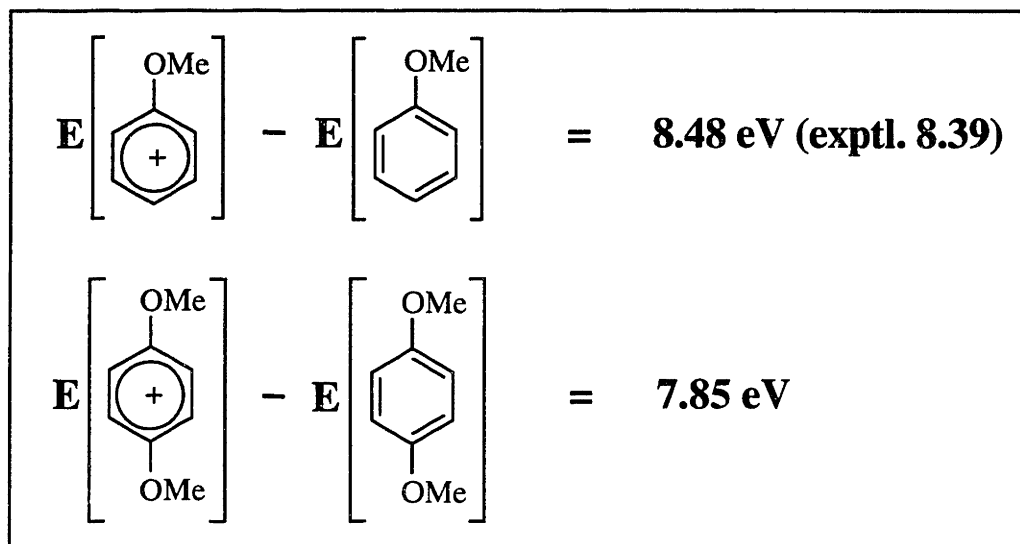


Figure A2.7. Computational Results for the Ionization Potential. Geometries are determined by geometry optimization at the AM1 level.

As shown in Figure A2.7, the additional methoxy group does not appear to sufficiently activate the aromatic substrate to produce high yields of DMB. *Ab initio* calculations at the 3-21G* level are only reliable, however, to within a few tenths of an electron volt. Thus, the computational results are not decisive. Furthermore, the quantitative criterion is valid only for the specific reaction conditions. By changing the reaction conditions to enhance the stability of the transition state relative to the reactant molecule, it may be possible to reduce the upper bound on the ionization potential.

Literature on Synthesis of DMB *via* Vilsmeier–Haack Formylation

The Vilsmeier–Haack formylation has been used several times in the literature to synthesize DMB, usually under the auspices of a proposed general procedure. The yields are generally below 20% for both dimethylformamide–POCl₃ and methylformanilide–POCl₃ systems. Relevant excerpts are included in the appendix.

A characteristic feature of the reported syntheses is the use of excess formamide as the solvent. Higher yields could almost certainly be obtained by solvent optimization, which consists mainly of optimizing a tradeoff between reagent solubilization and free energy of activation for the rate determining step. Solvent selection criteria have been presented (Linda *et al.*, 1974).

- Solvents of low polarity hinder the solvation of the formamide–acid chloride adduct. Thus, solvents such as benzene and hexane are not suitable;
- protic solvents will induce decomposition of certain adducts; also, formamides are not suitable solvents for certain combinations of reagents (*e.g.*, dimethylformamide–COCl₂);
- strongly polar solvents tend to stabilize the electrophilic formamide–acid chloride adduct relative to the transition state, thereby reducing the reaction rate; and
- successful solvents include chloroform, 1,2–dichloroethane, and acetonitrile.

Furthermore, a number of different acid chlorides are applicable, including COCl₂; SOCl₂; ClCOCOCl; CH₃COCl; ArCOCl; ArSO₂Cl; PCl₅; AlCl₃; MeNSO₂Cl; and RO₂CNHSO₂Cl. One chlorine–free reagent that has successfully extended the scope of the formylation by increasing leaving group mobility is the anhydride of trifluoromethanesulfonate, (CF₃SO₂)₂O.

Patent Citations

Recent Patents:

- Huet, M. and D. Nobel, "Process for the Preparation of Mono– or Polyalkoxylated Aromatic Compounds," *European Pat. Appl.* 455,545 A1 (to Rhone–Poulenc Chimie), November 6, 1991.
- Dulenko, V., N. Alekseev *et al.*, "2,5–Dimethoxybenzaldehyde," *USSR Pat.* 662,545 (to Institute of Physical–Organic Chemistry and Coal Chemistry, Academy of Sciences, Ukrainian S. S. R.), December 22, 1977.

Early Vilsmeier aldehyde synthesis patents:

- Kalischer, G. and K. Keller, *Germ. Pat.* 519,806 (1927).
- Kalischer, G., H. Scheyer, and K. Keller, *Germ. Pat.* 514,415 (1927).
- Wolff, P. *Germ. Pat.* 614,325; 615,130 (1933).

Appendix 3: Introduction to Systems Engineering

A3.1. Overview

Computer-aided modeling and engineering are now applied at a hierarchy of levels in process development. Starting in the initial stages of product development, databases are extensively used to obtain basic chemical information. Kinetic simulation coupled with design of model compound experiments is used to understand product controlling factors and to develop optimal control strategies. Mathematical analysis, simulation, and optimization techniques are making headway in flowsheet design. Decision models are now developed in many organizations for strategic planning involving allocation and scheduling of resources. A common motivation at each level is the need to better assimilate constraints and objectives from a higher level of the hierarchy—*e.g.*, to avoid environmental problems in the first place through better up-front engineering. This appendix presents an introduction to computer-aided process engineering tools. A section is devoted to each of the following:

- Sequential Process Simulation
- Dynamic Simulation
- Equation-Based Process Optimization
- Integration of Process Subsystems
- Flexibility Analysis
- Modeling Uncertainty: Probabilistic Analysis
- Empirical Model Building
- Scheduling

The first four topics are the "classical domain" of computer-aided systems engineering (although they are still the subject of much research). In one sense, these methods help to systematize traditional chemical engineering activities such as study-level design estimation and process optimization. The first four methods are covered by briefly reviewing applications and providing a list of relevant computer programs.

The remaining four topics, in contrast, use computational methods to derive entirely new forms of engineering analysis. Flexibility analysis, for example, involves a systematic characterization of the operational flexibility of a process. To motivate possible uses of these last four methods, applications are considered in somewhat more detail and specific design instances discussed.

A3.2. Sequential Process Simulation

Sequential process simulation includes computation of steady-state material and energy balances for a process flowsheet model. This model is generally specified down to the level of individual unit operations as well as material and energy constraints. The term "sequential" means that unit operations models are solved one-at-a-time rather than simultaneously. This approach allows detailed calculations to be performed on individual unit operations.

Applications of process simulation include

- study-level design estimation for proposed chemical processes;
- characterization of existing chemical processes (*e.g.*, material and energy balances) and process components (*e.g.*, rigorous simulation of fractionation columns);
- sequential process optimization; and
- evaluation of changes to an existing process.

In each application, process simulation is a good starting point for (*i*) testing waste reduction strategies, (*ii*) analyzing of process sensitivity with respect to inputs and process parameters, and (*iii*) building more detailed models. In particular, process simulation is useful for the systematic economic and environmental comparison of two or more candidate processes.

A process simulation code includes

- a set of unit operation modules;
- thermophysical properties and models databanks;
- a flowsheet specification compiler;
- tools for building and incorporating user-defined process models and data; and
- tools for flowsheet optimization.

Modern commercial codes are equipped with a graphical interface, which can considerably simplify assembly and simulation of process flowsheets. Rudimentary use of the interfaces can be learned within a few hours. Two major commercial codes are listed in Table A3.1.

Table A3.1. Two Widely Used Process Simulator Codes.

Code	Distributor	Comments
ASPEN PLUS	Aspen Tech., Inc.	Most widely used in chemical (but not petrochemical) industry. Platforms include mainframes, PCs, and most engineering workstations.
PROCESS	Simulation Sciences, Inc.	Databanks tailored for petrochemical industry. Platforms include PCs and most engineering workstations.

It is important to select a process simulation code from a stable vendor such as Aspen Technology or Simulation Sciences so as to insure good customer support and consistent software upgrades.

A3.3. Dynamic Simulation

Dynamic simulation is the simulation of time-dependent process models to determine a transient response surface. Whereas sequential process simulators treat unit operations separately, dynamic simulation is most often associated with equation-based modeling in which the entire flowsheet is represented by a single system of nonlinear, differential-algebraic equations.

Applications of dynamic simulation have grown rapidly with increases in computing power and related advances in numerical solution algorithms. Dynamic simulation is currently used to

- study dynamical response of process systems;
- assess violations of feasible operating ranges;
- identify possible safety risks; and
- train plant operators to minimize down-time and safety risks.

A real success has been the use of dynamic process simulation in plant operator training at DuPont. For example, the program has been used to train operator personnel responsible for starting up and running DuPont's new refrigerant plants. Minimum down-time records have been set during startup these difficult-to-operate chemical plants.

The current status of dynamic simulation codes is not unlike that of process simulators 10–15 years ago. Namely, while most codes are developed internally to meet specific requirements, there is a steadily growing market for general-purpose commercial products. In particular, OSHA's new guidelines require detailed safety analysis that could

be met, in part, through dynamic simulation. The predominant commercial code today is SPEEDUP (Aspen Technology, Inc.; Perkins and Sargent, 1982). This program solves a system of continuous time differential-algebraic equations. It should also be noted that considerable progress has also been made towards general purpose discrete dynamic simulation—*i.e.*, dynamic simulation incorporating discrete events such as safety interlocks and changes in operating states. A discrete dynamic simulation compiler named g-PROMS (general Process Modeling System) code is now available (Barton and Pantalides, 1993).

It is important to recognize that dynamic simulators still require more modeling expertise as compared to steady-state simulation. In particular, available computational algorithms face certain limitations. One unresolved issue is the "index problem," which is related to consistent initialization of differential-algebraic systems (Chung and Westerberg, 1990).

A3.4. Equation-Based Process Optimization

Equation-based process optimization is the use of mathematical programming techniques (discussed below) to merge the tasks of process modeling and process optimization. Two broad, competing issues dominate current thinking on equation-based optimization: (i) the integration of process subsystems to improve a measure of the economic value (Morgan, 1992); and (ii) the flexibility and robustness of chemical processes in the presence of uncertain or variable inputs and process parameters (Swaney and Grossmann, 1985a). Often a significant challenge is to find the optimal level of process integration while retaining flexibility for feasible operation.

A common form for a mathematical program is given by Eqs. 1–3.

$$\min_{x,z} \mu(x, z, u) \quad (1)$$

$$\text{s.t. } h(x, z, u) = 0 \quad (2)$$

$$g(x, z, u) \leq 0 \quad (3)$$

in which μ is an evaluation function to be minimized (*e.g.*, cost); h is the vector of equality constraints (*e.g.*, mass balances or equilibrium relations); and g is the vector of inequality constraints (*e.g.*, design specifications or physical limits in a process). Each vector is a function of the following variables: x is the vector of design variables defining the choice of process elements; z is the vector of manipulated variables in a process; and u is the vector of state variables defined to be of the same dimension as h .

A relatively small number of mathematical programming techniques form the basis for most applications: linear programming, nonlinear programming, mixed-integer linear

programming, and mixed-integer nonlinear programming. A list of applications to chemical engineering problems is given in Table A3.2.

Table A3.2. Chemical Engineering Applications of Mathematical Programming.

Form	Applications (form is not necessarily exclusive)
Linear Program	zero-wait scheduling (Birewar and Grossmann, 1989); process optimization of linearized models; network flow problems; resource allocation problems; stochastic optimization
Nonlinear Program	flowsheet optimization; data reconciliation and gross error detection (Kumar <i>et al.</i> , 1993); optimization of design and control system (Luyben and Floudas, 1993)
Mixed-Integer Linear Program	batch process design and scheduling; flexibility analysis; heat exchanger network synthesis subproblems; steam system integration
Mixed-Integer Nonlinear Program	general process synthesis problems (<i>e.g.</i> , heat exchanger network synthesis); discrete component design; kinetic model reduction

Commercial compilers are available which provide a common interface to each method. Two reasonably-priced compilers are GAMS (General Algebraic Modeling System; \$10-20,000 for a good implementation; Brooke *et al.*, 1988) and AMPL (Fourer, 1993). The algorithms available in GAMS are sufficient for many problems consisting of perhaps hundreds of nonlinear constraints, thousands of linear constraints, and up to several hundred discrete variables. It should be noted, however, that algorithm performance tends to be very problem dependent in integer optimization. Code recommendations are given in Tables A3.3-5.

Table A3.3. Codes and Recommendations for Linear Programming.

Code	Algorithm	Comments
OSL*	simplex, dual simplex, Karmakar, network flow algorithms	best all-around commercial code of the set
CPLEX**	simplex	high quality code, used to be the best
MINOS (Murtagh and Saunders, 1987)*	simplex	benchmark of performance
ZOOM*	simplex	benchmark of performance
LINDO	simplex	interactive; inexpensive

* Available in GAMS.

** Link to GAMS available.

*** Available in academic research version of GAMS (Carnegie Mellon University, 1990-1992).

Table A3.4. Codes and Recommendations for Constrained Nonlinear Programming.

Code	Algorithm	Comments
MINOS*	reduced gradient algorithm	benchmark of performance
RND-SQP (Vasantharajan <i>et al.</i> , 1990)***	range and nullspace decomposition / sequential quadratic programming	for problems with relatively few degrees of freedom; generates dense matrices

Table A3.5. Codes and Recommendations for Mixed-Integer Linear Programming.

Code	Algorithm	Comments
OSL*	branch-and-bound / branch-and-cut	best all-around commercial code of the set
CPLEX**	branch-and-bound	extension of LP code
SCICONIC**	branch-and-bound with limited cutting plane capabilities	poor LP solver but otherwise good code
ZOOM-XMP*	branch-and-bound; pivot-and-complement heuristic used	not as robust as OSL

* Available in GAMS.

** Link to GAMS available.

*** Available in academic research version of GAMS (Carnegie Mellon University, 1990-1992).

Two types of algorithms for mixed-integer nonlinear programming are mentioned here. The Generalized Bender's Decomposition has been applied to a number of different problems both inside and outside the engineering domain (Geoffrion, 1972). More recently, the mixed-integer nonlinear programming codes DICOPT (Kocis and Grossmann, 1987) and DICOPT+ (Viswanathan and Grossmann, 1990) have been implemented in GAMS. DICOPT+ is available commercially.

This discussion represents only a small fraction of the relevant information on mathematical optimization. Further codes incorporating specialized algorithms as well as mathematical programming formulations can be discussed if a particular need arises.

A3.5. Integration of Process Subsystems

Process integration is the use of equation-based, sequential, or combined approaches for integration of tightly coupled process subsystems. A number of tradeoffs can be present in integrated systems motivating computer-aided analysis. Typical examples of such tradeoffs are well-illustrated by the flowsheet in Figure A3.1, which is a simplified depiction of the process for ammonia manufacture (adapted from Lang *et al.*, 1988).

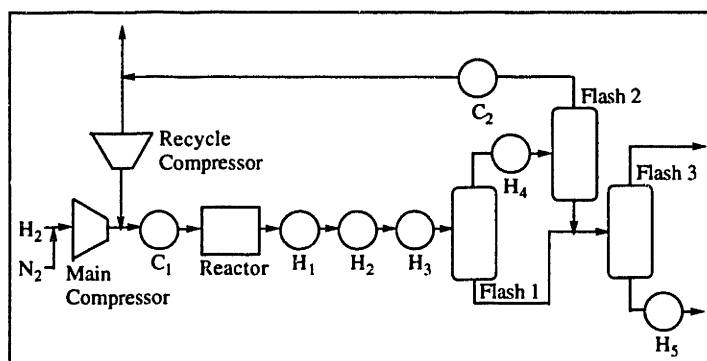


Figure A3.1. Simplified Illustration of Ammonia Process Flowsheet (Lang *et al.*, 1988). *H* and *C* are heaters and coolers respectively.

The feedstocks, nitrogen and hydrogen, are compressed and mixed with a hydrogen rich recycle stream. The feed stream undergoes an endothermic reaction and partial conversion to ammonia. The mixed product stream is heated in stages and undergoes several separations. Nitrogen is purged in the third flash while product ammonia is taken as bottom product. In this flowsheet, there is an interdependency between the compressor work and the efficiency of the flash units. Better separations can be achieved by increasing pressure drops in the compressors, but increasing the pressure drops will also increase the work of compression. There are also tradeoffs between energy efficiency and capital expense. In particular, the heating and cooling streams can be integrated to reduce utilities costs, but this reduction must be balanced against potentially higher capital costs.

Mathematical programming techniques, pinch technology (Rossiter *et al.*, 1993), and hybrid approaches have been developed for heat integration. An example of a heat integrated network is shown in Figure A3.2 (Lang *et al.*, 1988). The configuration has been chosen to optimize a measure of the net present value of the process by making the least expensive tradeoff between capital investment in heat exchangers and the costs of utilities over time.

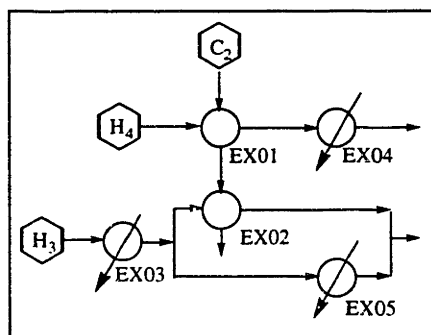


Figure A3.2. Portion of the Optimized Heat Exchanger Network Developed for Ammonia Process (Lang *et al.*, 1988). *EX* denotes a heat exchanger.

Process integration is by no means restricted to heat exchanger networks. Andreovich and Westerberg (1985) have developed an approach for the synthesis of heat-integrated distillation sequences. Grossmann and Santibanez (1980) have considered mixed-integer linear programming strategies for steam system integration. El-Halwagi and Manousiouthakis (1990) have studied synthesis of mass-exchange and regeneration networks. El-Halwagi (1992) has considered synthesis of reverse-osmosis networks for waste reduction. El-Halwagi and Srinivas (1992) have studied synthesis of reactive mass exchange networks.

A3.6. Flexibility Analysis

One of the problems with process integration is a potential reduction in process flexibility, inducing potential control limitations and safety hazards. Flexibility analysis is the use of mathematical programming techniques to characterize the operational flexibility of a process (Swaney and Grossmann, 1985a,b; Grossmann and Floudas, 1987; Straub and Grossmann, 1990 and 1992). Such analysis has a number of applications, including

- identification of critical design assumptions in integrated systems;
- risk analysis (*e.g.*, probability of failure);
- identification of parameters whose variance should be reduced; and
- comparison of flexibility of alternative designs.

A detailed presentation of the methods is beyond the scope of this report, but the basic idea can be illustrated with a simple example of a pump and pipe run (Figure A3.3; Swaney and Grossmann, 1985a).

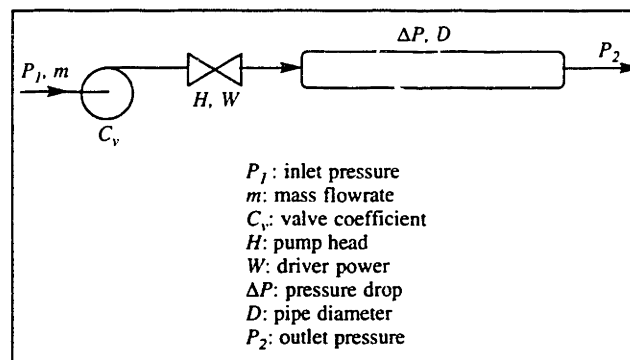


Figure A3.3. Simple Pump and Pipe Run for Flexibility Analysis Example.
Adapted from Swaney and Grossmann (1985a).

Given a set of uncertain or fluctuating model parameters, the basic goal in [deterministic] flexibility analysis is to find the range of the parameters over which control variables can be manipulated so that operating constraints are satisfied. In the present

example (Swaney and Grossmann, 1985a), the uncertain parameters are the mass flowrate, m , the desired pressure, P_2^* . The control variable is the value coefficient, C_v . The two uncertain parameters have ranges $(m - \Delta m^-, m + \Delta m^+)$ and $(P_2^* - \Delta P_2^{*-}, P_2^* + \Delta P_2^{*+})$. Operating constraints consist of the mechanical energy balance (Eq. 4), outlet pressure tolerance (Eq. 5), pump driver power limit (Eq. 6), and the control valve range (Eqs. 7 and 8).

$$P_1 + \rho H - \frac{m^2}{\rho C_v^2} - k m^{1.84} D^{-5.16} - P_2 = 0 \quad (4)$$

$$P_2^* - \varepsilon \leq P_2 \leq P_2^* + \varepsilon \quad (5)$$

$$\frac{mH}{\eta} \leq W \quad (6)$$

$$r C_v^{MAX} \leq C_v \quad (7)$$

$$C_v \leq C_v^{MAX} \quad (8)$$

In these equations, the following constants are defined: k characterizes the pressure drop in the pipe; r defines the control valve range; η is the pump efficiency; and ρ is the liquid density. Substitution of Eq. 4 into Eq. 5 yields Eqs. 9–10.

$$\left(P_1 + \rho H - \frac{m^2}{\rho C_v^2} - k m^{1.84} D^{-5.16} \right) - P_2^* - \varepsilon \leq 0 \quad (9)$$

$$-\left(P_1 + \rho H - \frac{m^2}{\rho C_v^2} - k m^{1.84} D^{-5.16} \right) + P_2^* - \varepsilon \leq 0 \quad (10)$$

The flexibility of the system can be characterized by a flexibility index, $\delta \in [0, 1]$, substituted into the parameter ranges, namely Eqs. 11 and 12.

$$(m - \delta \Delta m^-, m + \delta \Delta m^+) \quad (11)$$

$$(P_2^* - \delta \Delta P_2^{*-}, P_2^* + \delta \Delta P_2^{*+}) \quad (12)$$

One of the goals of flexibility analysis is to find the largest value of δ , denoted δ^* , for which Eqs. 11 and 12 satisfy all of the operational constraints. A convenient way of viewing the problem is to plot the operational constraints (with the state variables substituted out) and the rectangle (Eqs. 11 and 12) generated around the nominal values for the uncertain parameters (Figure A3.4).

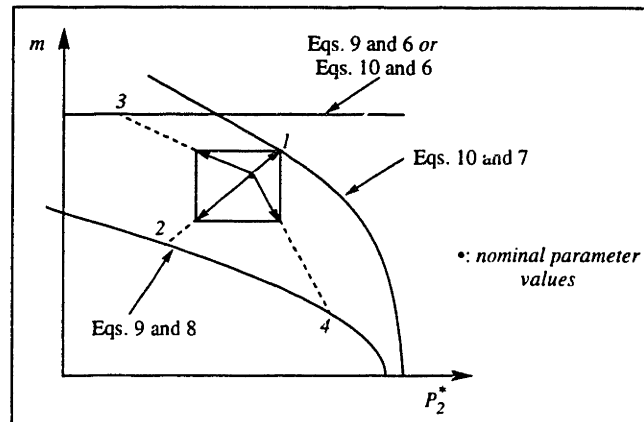


Figure A3.4. Determination of the Flexibility Index and the First Violated Constraint. The numbers indicate the order in which constraints are violated. Adapted from (Swaney and Grossmann, 1985a).

The critical operational constraint is that defined by Eqs. 7 and 10, since it comes in contact with the rectangle first as δ is increased. The order of contact for the remaining constraints is also indicated in Figure A3.4. The first contact occurs at a point defined by $\delta^* < 1$. Under the assumptions of the analysis, in particular that the parameter uncertainties are uniformly distributed, the design flexibility can be increased only by moving the critical constraint away from the nominal parameter values. This might be accomplished, for example, by increasing the pump head, ρH . Proceeding in a systematic fashion through a succession of designs, a design that exactly satisfies the flexibility requirements (*i.e.*, $\delta^* = 1$) can be identified. Whereas the analysis in Figure A3.4 is graphical, consideration of problems with many parameters and constraints involves the use of discrete optimization techniques described in the references given above.

Additional forms of flexibility analysis have been developed. Stochastic flexibility analysis, for example, uses the density function of an uncertain parameter to compute the statistical probability of feasible operation (Straub and Grossmann, 1990; 1992). Recently, stochastic flexibility analysis has been applied to optimal batch process design and scheduling in the presence of parametric uncertainty (Straub and Grossmann, 1992). It should be noted that available algorithms for flexibility analysis are sensitive to the form of nonlinearities; the methods can fail for general nonconvex constraints.

A3.7. Modeling Uncertainty: Probabilistic Analysis

Most systems encountered in chemical engineering are in some way subject to modeling uncertainty as well as changing or uncertain inputs and demands. Flexibility analysis was discussed above as a tool for studying the operational range of a design as a function of model parameters. At a more detailed level, the actual distribution of inputs and outputs is of considerable importance as well. Probabilistic analysis is one of the techniques available to characterize the distribution of system variables given fluctuations or uncertainty in other variables or parameters (Figure A3.5).

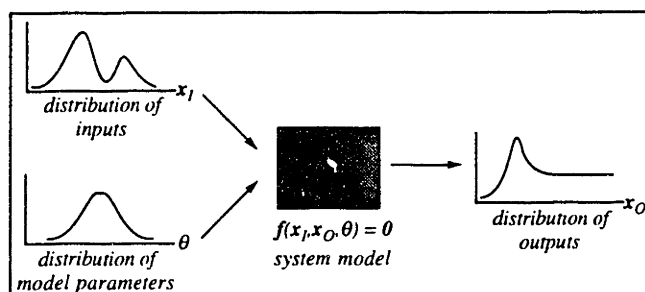


Figure A3.5. Probabilistic Treatment of Input and Model Uncertainties to Determine Output Distribution.

Outputs, which are denoted x_O , are not restricted to explicit outputs of the process. They can be any choice of variables that fluctuates in response to a separate set of variables, x_I , or parameters, θ . The set of equations denoted by $f(\bullet)$ is general. In the case of an optimal control problem, $f(\bullet)$ is a differential-algebraic system. In a nonlinear-algebraic optimization problem, $f(\bullet)$ is the set of stationary conditions and active constraints.

Relevant applications of uncertainty analysis include the following:

Application 1. Determine steady-state inputs and process parameters for which uncertainty should be reduced. The uncertainty can be reduced through further experimental measurements or through overdesign (Figure A3.6). Examples of such parameters, drawn from different design problems, include (i) uncertainty in future airborne emissions requirements; (ii) evolution of catalyst particle size distribution in a reactor; and (iii) variance in feed composition to a petrochemical facility. Figure A3.6 illustrates that a process design yielding the highest NPV (arbitrary economic measure) at a nominal parameter value, $\bar{\theta}$, may be sensitive to fluctuations, $\delta\theta$, in that parameter.

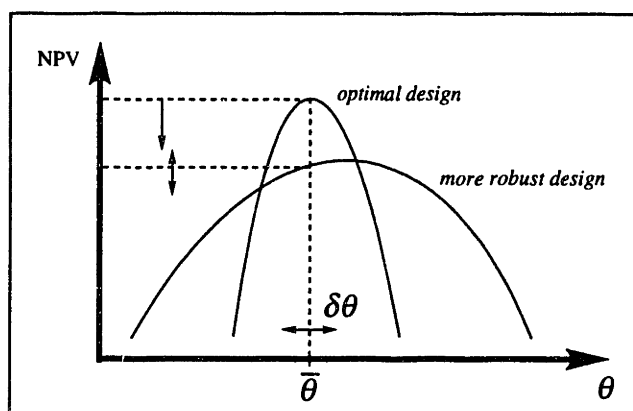


Figure A3.6. Robustness of the Net Present Value of a Process to Changes in an Input Parameter θ . NPV denotes an economic measure.

Modification of the evaluation function to simultaneously account for a range parameter values can lead to a more robust design that is less sensitive to uncertainty or fluctuation in the parameter(s).

Several different approaches are available for evaluating chemical processes over a range of parameters. Monte Carlo methods can be used to sample from a multivariate distribution of inputs. An important advantage of Monte Carlo methods is that they can be used with sequential process simulation whether or not an equation-based representation is available. Such methods, however, can be quite time consuming when multiple uncertain parameters are considered.

When an explicit equation-based representation is available, so-called direct methods can be exploited. In direct methods, an uncertain parameter is expanded in random variable basis functions. Ghanem and Spanos (1991) showed that any square-integrable random variable, $x(\omega)$, can be expanded in polynomial functionals, $\Xi_i[\bullet]$, of Gaussian distributed random variables, $\xi_j(\omega)$ (depicted qualitatively in Eq. 13).

$$x(\omega) = \sum_i c_i \Xi_i \left[\left\{ \xi_{j(i)}(\omega) \right\} \right] \quad (13)$$

Equation 13 is known as the polynomial chaos expansion. These methods have several advantages over Monte Carlo methods in certain cases (Tatang, 1994). First, the method is equivalent to variational schemes such as finite element methods. Application of variational principles allows conversion of the original problem to a deterministic form that can be solved with existing matrix solvers. Included in the problem solution are the coefficients c_i , from which the full distributional properties of the output variables can be determined. Second, the functional can often be orthogonalized to eliminate cross terms, resulting in a formulation that is computationally amenable to very large-scale problems.

Third, the method can dramatically reduce the computational expense versus Monte Carlo methods if the number of terms in the expansion can be kept to a relatively small number (e.g., < 5). It should be noted that compilers for conversion of deterministic models using polynomial chaos expansions are currently under development (Tatang, 1994).

Application 2. A task complementary to application 1 is the determination of optimal oversize factors. Powers *et al.* (1975) considered design of a catalytic reactor-separator system allowing for uncertainty in the catalyst activity and selectivity. The capital cost of a process system can be treated as $c[\mathbf{d}, \omega]$, where \mathbf{d} is a vector of oversize factors and ω is a vector of uncertain parameters with a probability measure $P[\omega]$. Choosing a small oversize factor can reduce nominal capital costs but increases the range of ω (and hence $P[\omega]$) for which the process is underdesigned—*i.e.*, requiring additional capital to rebuild the reactor and separator. Thus there is a tradeoff between nominal capital expenses and the probability of having to later rebuild (Figure A3.7).

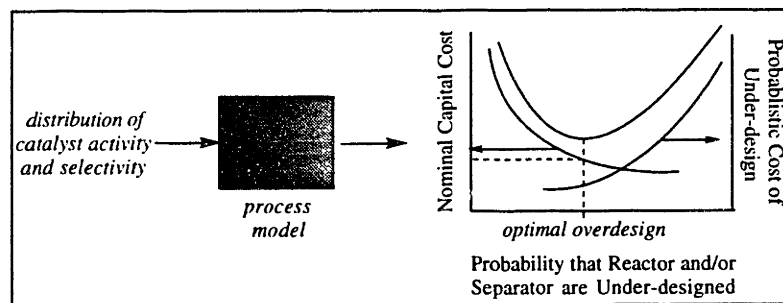


Figure A3.7. Uncertainty in Catalyst Activity and Selectivity Creates a Tradeoff in Oversize Costs Versus Expected Cost of Rebuilding Reactor and Separator.

As indicated in Figure A3.7, the optimum oversize can be treated as the choice of oversize factors to minimize the expected cost (Eq. 14).

$$\bar{c} = \min_{\mathbf{d}} \int_{\Omega} c[\mathbf{d}, \omega] dP(\omega) \quad (14)$$

Application 3. Optimal design in the presence of uncertainty [stochastic optimization]. Applicable techniques build upon mathematical programming and probabilistic modeling tools. Stochastic optimization problems can be posed by augmenting a deterministic mathematical program with a vector of uncertain parameters, $\theta \in \Theta$. Two possible forms of the design objective are given in Eqs. 15 and 16.

$$\int_{\Theta} \min_{x,z} \mu(x, z, u, \theta) dP(\theta) \Rightarrow x^{optimal}(\theta), z^{optimal}(\theta), u^{optimal}(\theta) \quad (15)$$

$$\min_{x,z} \int_{\Theta} f[\mu, g, h(x, z, u, \theta)] dP(\theta) \Rightarrow x^{optimal}, z^{optimal}, u^{optimal} \tag{16}$$

Equation 15 essentially treats all of the model variables as random variables. The result is the statistical distribution of optimal designs for $\theta \in \Theta$. Applications of this approach include

- identifying which design and control variables are most sensitive to parametric uncertainty at the point of optimality;
- characterizing the behavior of the objective function in the space $\theta \in \Theta$; and
- constructing a tradeoff curve between the choice of designs as a function of $\theta \in \Theta$.

Monte Carlo sampling from $\theta \in \Theta$ followed by deterministic optimizations is the most generally applicable method for solving Eq. 15. Alternately, Ciric and Jia (1994) have proposed a mathematical programming method in which the distribution is approximated by piecewise-convex polytopes.

One of the limitations of Eq. 15 is that it does not yield a single design that can be implemented. Equation 16, on the other hand, finds a single design that minimizes an objective function averaged by the parametric uncertainty. In this averaging it is possible to incorporate constraints that increase process robustness in the presence of uncertainty. A compendium of commonly encountered stochastic programs is given in Eqs. 17–20.

$$\min_{x,z} \int_{\Theta} \left\{ \sum_{i=1}^N c_i [\mu(x, z, u, \theta) - \bar{\mu}]^i + w^T \alpha + y^T \beta \right\} dP(\theta) \tag{17}$$

$$P[-\varepsilon \leq h(x, z, u, \theta) \leq \varepsilon] \geq \beta \tag{18}$$

$$P[g(x, z, u, \theta) \leq \theta] \geq \alpha \tag{19}$$

$$[\mu(x, z, u, \theta) - \bar{\mu}]^2 \leq \text{Upper Bound}, \theta \in \Theta \tag{20}$$

$$\mu(x, z, u, \theta) \leq \text{Upper Bound}, \theta \in \Theta \tag{21}$$

where $\bar{\mu}$ is determined from the deterministic optimization at the nominal parameter value, $\bar{\theta}$.

The objective function given by Eq. 17 incorporates three main terms. The first term is a weighted summation over moments of the economic function. For example: $[\mu(x, z, u, \theta) - \bar{\mu}]$ integrates to give the average (*i.e.*, expected) value of the objective

function; $[\mu(x, z, u, \theta) - \bar{\mu}]^2$ integrates to give the variance. Thus a positive weighting on the latter term incorporates a willingness to accept a higher process cost in order to reduce the variance in that cost. This idea is illustrated in Figure A3.8.

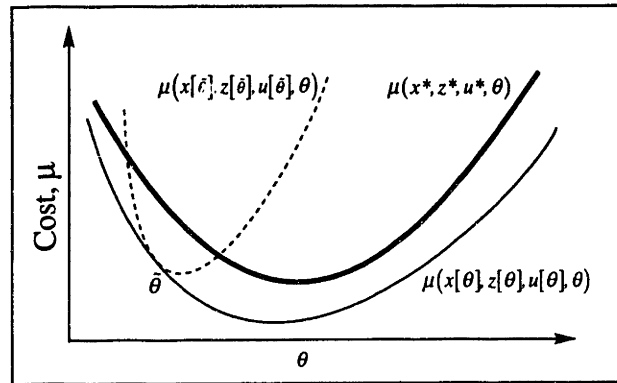


Figure A3.8. Illustration of Tradeoff Between Optimal Design for a Single $\tilde{\theta} \in \Theta$ Versus Choice of a Design $\langle x^*, z^* \rangle$ that is Less Sensitive to $\theta \in \Theta$.

Although the result obtained from Eq. 15 (i.e., $\mu(x[\theta], z[\theta], u[\theta], \theta)$) will always be a lower bound on the cost, an optimal design with respect to a single $\tilde{\theta} \in \Theta$, i.e., $\mu(x[\tilde{\theta}], z[\tilde{\theta}], u[\tilde{\theta}], \theta)$, may become costlier for other $\theta \in \Theta$ (Figure A3.8). Incorporation of the variance into the objective function tends to yield a design $\langle x^*, z^* \rangle$ that is less sensitive to variation in $\theta \in \Theta$.

The second and third terms in the objective function are derived from Eqs. 18 and 19. It is important to recognize that in any given design, Eqs. 2 and 3 are likely to be violated for a subset of $\theta \in \Theta$. In order to control the degree to which the constraints are violated, Eqs. 18 and 19 admit that Eqs. 2 and 3 can be violated with a probabilities β and α , but that costs $y^T \beta$ and $w^T \alpha$ will be incurred in the objective function as a result. Thus these constraints, known as chance constraints (Taha, 1987), assign a cost to violation of Eqs. 2 and 3.

Equation 20 is a hard constraint on the variance that can be used in place of the second moment in the objective function. Similarly, Eq. 21 is a hard upper bound on the cost function that must be satisfied at every $\theta \in \Theta$. Equation 21 is often used as a constraint in the optimization of hedged fund portfolios when the design objective is to minimize the variance for a target rate of return.

Whereas this discussion has considered general problem forms, it should be noted that the actual formulation of stochastic optimization problems tends to highly specialized to the problem of interest. Furthermore, special solution strategies (e.g., constraint aggregation-disaggregation schemes) and algorithms (e.g., tailored simplex algorithms) are

often required due to large problem sizes, particularly when the constraints represent a multistage optimization problem (Clay, 1994).

Application 4. Analysis of resource allocation, production planning, and scheduling decisions in the presence of stochastic market demand, short-term cost of capital, and other phenomena. My experience with stochastic modeling involves analysis and optimization of mortgage derivative portfolios. Stochastic differential equations are used to model, among other things, short and long term interest rates. The distributions of asset and liability prices can be evaluated along each interest rate path to determine their mean value and variance over time. Such information facilitates optimal selection of assets and issuance of liabilities, selection of an appropriate risk level, and construction of a hedging strategy. It is interesting to note that within Exxon and other petrochemical companies, stochastic modeling (with techniques described below) is used in the oil futures market.

Practical computational techniques for stochastic modeling of time series originate largely from the financial sector. Two relevant and interrelated tools are stochastic differential equations and lattice diffusion models (Kerestecioglu, 1992). A stochastic differential equation (Eq 22) expresses the differential change, $dx(x, \tau)$, in terms of drift, $\mu_x(\bullet)$, volatility, $\sigma_x(\bullet)$, and stochastic shocks, $d\omega(\tau)$.

$$dx(x, \tau) = \mu_x(x, \tau)d\tau + \sigma_x(x, \tau)d\omega(\tau) \quad (22)$$

Stochastic differential equations can be solved by hybrid numerical integration-Monte Carlo methods (Greiner *et al.*, 1988) and by statistical fitting to a discretized lattice (Kerestecioglu, 1992). A lattice diffusion model discretizes the stochastic differential equation into its temporal and spacial (*e.g.*, short term interest rates) components (Figure A3.9).

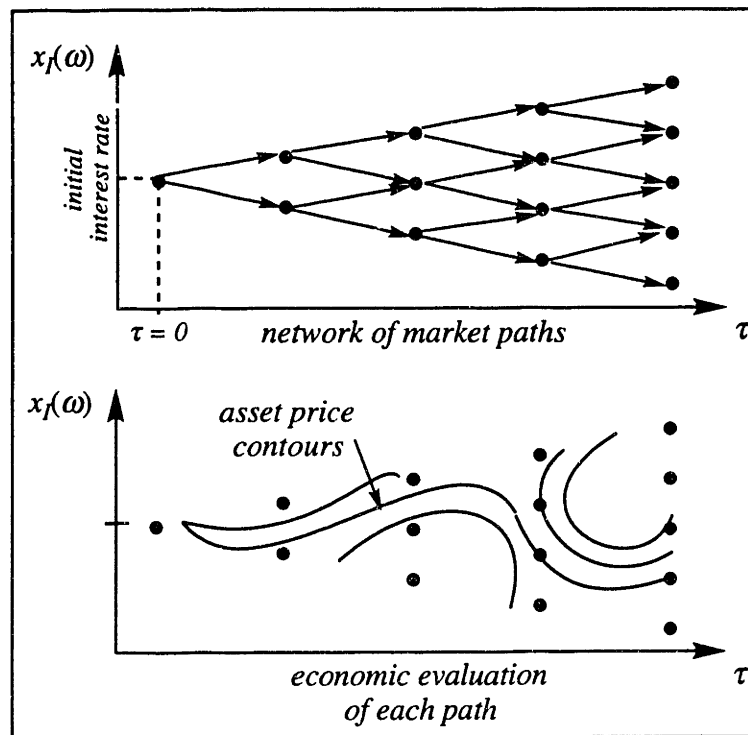


Figure A3.9. Single Factor Binomial Lattice Used for Generation of Interest Rate Scenarios According to a Stochastic Differential Equation. The lattice also provides a convenient representation of computational results, in this case consisting of asset prices over the scenarios.

An advantage of the lattice approach is that the parameters in the stochastic differential equation can be dynamically updated through multivariate regression of new market data incorporated into the lattice.

Probabilistic analysis has been offered here as an example of one approach to modeling uncertainty. Other approaches used in chemical engineering include interval analysis (Moore, 1966), fuzzy set theory (Zadah, 1965), and Dempster-Shafer belief theory (Shafer, 1976).

A3.8. Empirical Model Building

Empirical model building encompasses local and functional approximation methods useful in the absence of an explicit system model. Empirical modeling techniques can be broken into the several categories depicted in Figure A3.10.

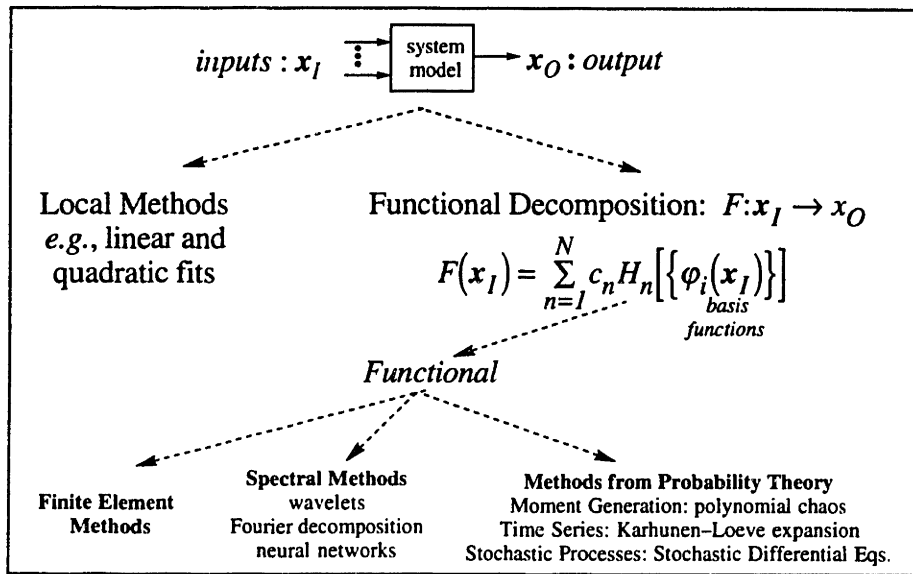


Figure A3.10. Empirical Model Building Methods for Characterization of Steady-State and Transient Systems.

A number of reasons may lead to development of empirical models. (i) There are important process systems where quantitatively accurate dynamic models are not available, including certain reaction-crystallization systems and three-phase polymerization reactors. (ii) Systems can exhibit a combination of short time-scale effects and noise that significantly hinders model verification. (iii) Process output can change unpredictably over time due to factors such as heat exchanger fouling. (iv) Mechanistic models of complex phenomena can involve a large number of variables and adjustable parameters. Applications of empirical modeling include

- incorporation of process measurements in a periodically updated empirical model (e.g., in model predictive control);
- separation of noise from the underlying model;
- spectral analysis of response surfaces for complex systems; and
- data compression and model aggregation.

Local methods are useful for characterization of steady-state operation around a single set of inputs. Functional decomposition methods are used for more complex situations. Often it is of interest to predict an entire response surface for a range of input values, x_I . Applications include model predictive control (ref. needed here) and equipment scaleup calculations. Finite element methods (or spline fitting methods) decompose the response surface into compact elements. This approach allows accurate fitting with a relatively small number of basis functions in any one element, although a

large number of elements may be required. Spectral approaches, on the other hand, attempt to approximate the entire response surface by global basis functions. In general, the spectral approaches involve a considerably larger number of basis functions.

One of the goals of experimental model building is the prediction of a response surface for achieving the optimum of a chemical process, either in design or in model predictive control. The response surface can be either time-independent or time dependent, depending on the nature of the system under consideration. A traditional difficulty in achieving the response surface has been the need to perform a considerable number of experimental measurements. The experimental requirements can be very demanding, particularly when the response surface is transient or when response surfaces are needed for a number of different design instances. Empirical techniques can potentially reduce the number of experiments through the construction of optimal basis functions for a more efficient spectral representation. From a knowledge of the basis functions, an approximate response surface can be generated dynamically from a much smaller number of experimental measurements.

An important category of the spectral techniques which allow generation of empirical basis functions arise from the probability theory of random processes. One such theory is the Karhunen-Loève expansion of a time-dependent response surface (Eq. 23; originally developed for random processes; Tatang, 1994):

$$F(x_I, t) = \sum_{n=1}^{\infty} c_n \alpha_n(t) \beta_n(x_I). \quad (23)$$

The method can be applied in discrete form based on a series of measurements taken at distinct time intervals. In particular, optimal empirical eigenfunctions can be generated that account both for time-dependent and input-dependent correlation in the output. The basis functions for the input-dependent correlation can be saved and reused for construction of a new response surface. Construction of this response surface will involve a much smaller number of measurements, taken at an instant in time. To illustrate, consider the set of transient response surfaces depicted in Figure A3.11. Each curve is constructed from fifteen values of the independent variable.

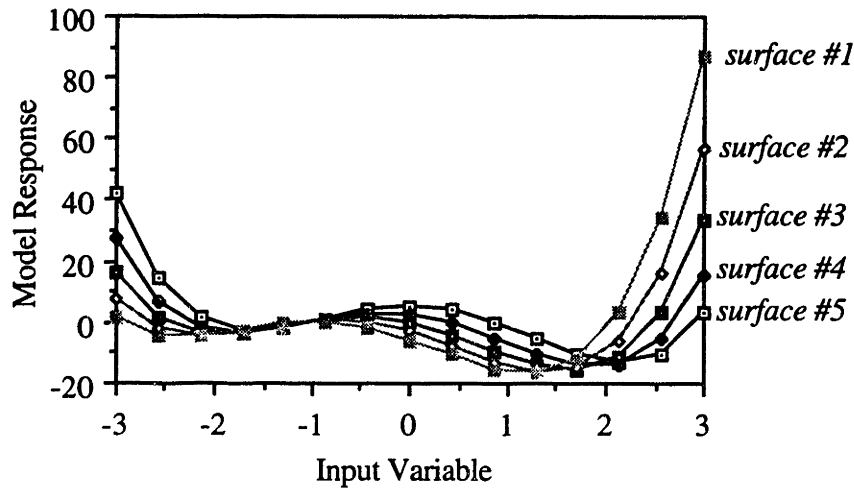


Figure A3.11. Transient Response Surfaces for the Example Problem

Application of the Karhunen-Loève decomposition yields a set of five orthogonal eigenfunctions of the input variable and corresponding eigenvalues. The eigenfunctions associated with the three largest eigenvalues are given in Figure A3.12.

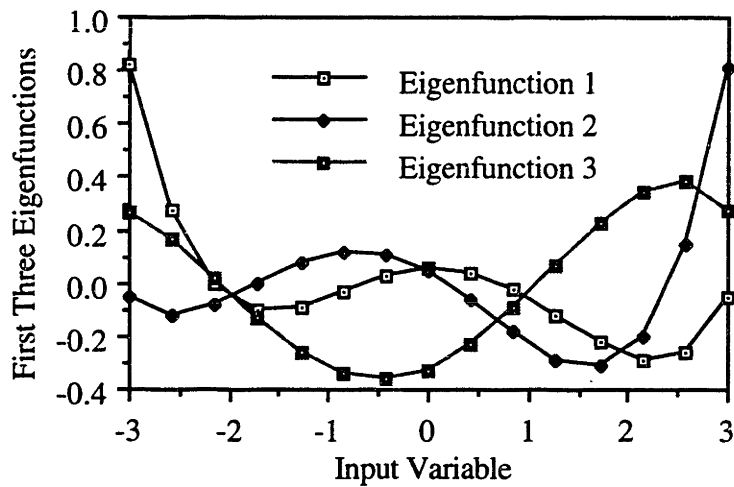


Figure A3.12. First Three Orthogonal Eigenfunctions Computed from Transient Response Surfaces (Figure A3.11).

These eigenfunctions can be used as basis functions to construct a response surface from a small number of new measurements. Figure A3.13 illustrates a fit to a measured response surface not included in the original dataset. Only the first two eigenfunctions are

utilized and only four points from the measured response surface are supplied. The coefficient of determination for the fit is better than 0.999.

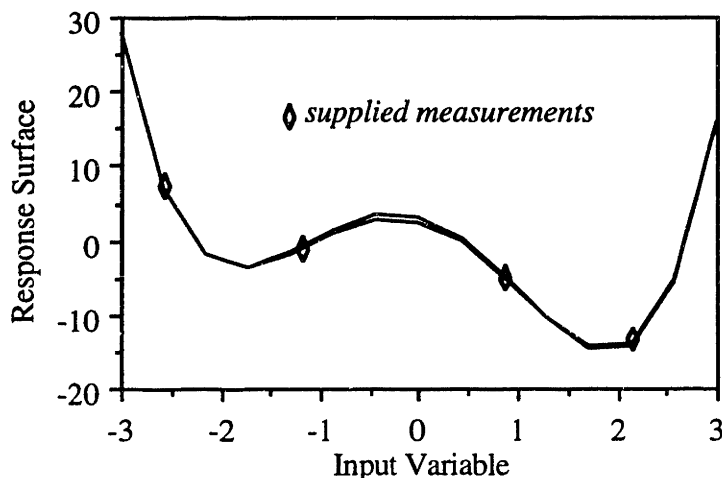


Figure A3.13. Approximation of a New Observed Response to the First Two Eigenfunctions (Figure A3.12). Measurements from four points are used to obtain the fits.

The Karhunen-Loève approach could be very useful for model predictive control. For example, in the case of a complex batch reactor, a comprehensive set of experiments could be used to construct the eigenfunctions for the temperature profile as a function of reactor inputs. Given the set of eigenfunctions, a much smaller number of periodic measurements could be used to update the response surface supplied to a controller.

Development of empirical modeling applications is generally done in-house. The underlying computational tools (such as codes for singular value decomposition and other matrix methods) are available in the public domain through NETLIB, or in commercial packages such as IMSL.

A3.9. Scheduling

Scheduling consists of heuristic (rule-based) and mathematical programming techniques for optimally sequencing a series of events and/or matching a set of tasks to a smaller number of available units. Chemical process engineering and operation can involve a number of scheduling problems. Several examples are described here.

Example 1. Given a set of geographically distributed crude oil sources and refineries, determine the minimum cost transportation network that meets inventory constraints. This type of problem involves a classic mathematical problem known as the Traveling Salesman Problem, in which the route between each geographic location is

weighted by a factor that represents travel distance and other costs. The graphical depiction of a Symmetric Traveling Salesman Problem (*i.e.*, the weighting from point 1 to point 2 is the same as from point 2 to point 1) is given in Figure A3.14.

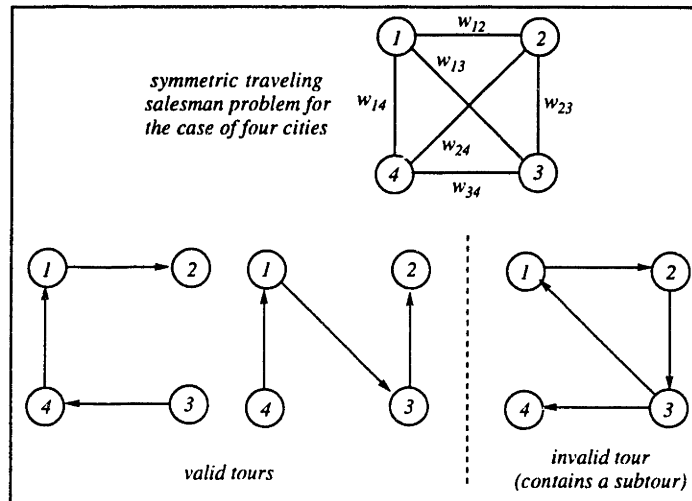


Figure A3.14. Graphical Depiction of Symmetric Traveling Salesman Problem for Four Cities.

Solution of large Traveling Salesman Problems has been an area of active research in telecommunications systems, routing problems, and chemical engineering problems in recent years. DuPont uses very large Traveling Salesman Problems (say 100,000 cities) in scheduling of chemical refineries (Miller and Pekny, 1991).

Example 2. Given a multipurpose plant for manufacturing of different colored polystyrene grades, choose the optimal sequence of colors (generally one wants to start with light colors and proceed towards dark colors) to minimize equipment cleanup time between individual batches. (Eastman Chemical Company is currently working this type of problem for dye-related compounds).

Example 3. Optimal scheduling and/or design of batch processes. Commonly encountered types of processes are listed in Figure A3.15.

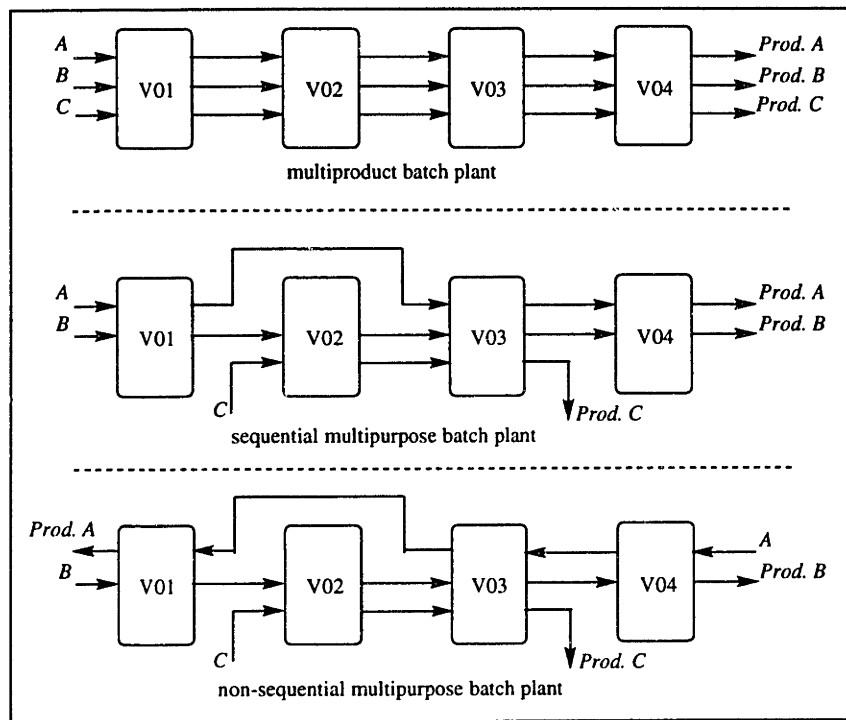


Figure A3.15. Different Types of Batch Scheduling and Design Problems. Adapted from Voudouris and Grossmann (1993).

Appendix 4: General Algebraic Modeling System (GAMS). Example Problem Specification and Results.

A4.1. Deterministic Case Study: Nominal Intermediate Prices

GAMS 2.25.056 VAX/VMS
simple example of aggregate task synthesis

05/15/93 17:24:28 PAGE 1

```
3 *
4 * coded by jonathan knight / 5-6-93
5 * dept. chem. engng.
6 * MIT
7 *
8 * COSTS ARE IN 1.0E4 DOLLARS.
9 * UTILITY COSTS ARE IN DOLLARS PER POUND MOLE AND ARE SCALED
10 * UP WITHIN THE NET PRESENT VALUE FACTOR
11
12 *
13
14 sets i /1*9/, j /1*3/, k /1*40/, m(k) /1*40/, rxn /1*4/, l /1*20/;
15
16 sets IJ(i,j)
17 /
18 1.1*3
19 2.1
20 2.3
21 3*4.1*2
22 5.1*3
23 6.1*2
24 7.1*2
25 8.1*3
26 9.1*2
27 /;
28
29 * --> SUBSET OF IJ FOR WHICH ADDITIONAL SEPARATION IS AN OPTION
30 set SEPX(i,j)
31 /
32 4*5.1
33 7*8.1
34 1.2
35 /;
36
37 * --> SET OF FEED STREAMS
38 set K0(i,j,k)
39 /
40 1*3.1.1
41 4*6.1.2
42 7*8.1.3
43 9.1.4
44
45 1.2.5
46 3*4.2.6
47 5.2.7
48 6.2.11
49 7.2.7
50 8*9.2.8
51
52 1.3.9
53 2.3.10
54 5.3.12
55 8.3.13
56 /;
57
58 * --> SET OF PRIMARY PRODUCT STREAMS
```

```
59 set K1(i,j,k)
60 /
61 1*2.1.5
62 3*4.1.6
63 5.1.7
```

GAMS 2.25.056 VAX/VMS
simple example of aggregate task synthesis

05/15/93 17:24:28 PAGE 2

```
64 6.1.11
65 7.1.7
66 8*9.1.8
67
68 1.2.9
69 3*4.2.10
70 5.2.12
71 6.2.16
72 7.2.12
73 8*9.2.13
74
75 1.3.14
76 2.3.15
77 5.3.17
78 8.3.18
79 /;
80
81 * --> SET OF SECONDARY PRODUCT STREAMS (REQUIRES ADDITIONAL SEPARATIONS)
82 set K2(i,j,k)
83 /
84 4.1.7
85 5.1.6
86 7.1.8
87 8.1.7
88
89 1.2.10
90 /;
91
92 * --> WASTE STREAM IF ADDITIONAL PURIFICATION IS UNDERTAKEN
93 set K3(i,j,k)
94 /
95 4.1.26
96 5.1.26
97 7.1.32
98 8.1.32
99
100 1.2.20
101 /;
102
103 * --> SET OF WASTE STREAMS ASSUMING NO ADDITIONAL SEPARATIONS
104 set K4(i,j,k)
105 /
106 1*2.1.19
107 3.1.24
108 4.1.27
109 5.1.39
110 6.1.30
111 7.1.33
112 8.1.40
113 9.1.37
114
115 1.2.21
116 3*4.2.25
117 5.2.28
118 6.2.31
119 7.2.34
120 8*9.2.35
121
122 1.3.22
123 2.3.23
124 5.3.29
```

GAMS 2.25.056 VAX/VMS
simple example of aggregate task synthesis

05/15/93 17:24:28 PAGE 3

```
125 8.3.36
126 /;
127
128 set FEED(k)
129 /
130 1*40
131 /;
132
133 set PRDD(k)
134 /
135 1*40
136 /;
137
138 * --> SET OF PRODUCTS THAT CAN BE SOLD
139 set PRDD2(k)
140 /
141 1*4
142 6*8
143 14*18
144 /;
145
146 set WASTE(k)
147 /
148 19*40
149 /;
150
151 * DISCRETIZATION SET FOR THE LAMBDA TECHNIQUE
152 sets q /1*3/, q1(q) /1*3/, q2(q) /1*3/;
153 sets s /1*2/, s1(s) /1*2/, s2(s) /1*2/;
154
155 parameter phi(i,j,k,q)
156 /
157 1*3.1.1.1 0.5
158 4*6.1.2.1 0.5
159 7*8.1.3.1 0.5
160 9.1.4.1 0.5
161
162 1.2.5.1 0.5
163 3*4.2.6.1 0.5
164 5.2.7.1 0.5
165 6.2.11.1 6.35295
166 7.2.7.1 0.5
167 8*9.2.8.1 0.5
168
169 1.3.9.1 4.9412
170 2.3.10.1 4.9412
171 5.3.12.1 4.9412
172 8.3.13.1 4.9412
173
174 1*3.1.1.2 3.0
175 4*6.1.2.2 3.0
176 7*8.1.3.2 3.0
177 9.1.4.2 3.0
178
179 1.2.5.2 3.0
180 3.2.6.2 3.0
181 4.2.6.2 0.75
182 5.2.7.2 3.0
183 6.2.11.2 7.0
184 7.2.7.2 0.75
185 8.2.8.2 3.0
```

GAMS 2.25.056 VAX/VMS
simple example of aggregate task synthesis

05/15/93 17:24:28 PAGE 4

```
186 9.2.8.2 0.75
187
188 1.3.9.2 7.0
189 2.3.10.2 7.0
190 5.3.12.2 7.0
191 8.3.13.2 7.0
192
193 1*2.1.1.3 11.11
194 3.1.1.3 17.12
195 4*5.1.2.3 15.77
196 6.1.2.3 13.5
197 7.1.3.3 17.12
198 8.1.3.3 15.77
199 9.1.4.3 15.2507
200
201 1.2.5.3 11.0914
202 3.2.6.3 11.11
203 4.2.6.3 1.243
204 5.2.7.3 11.11
205 6.2.11.3 12.70589
206 7.2.7.3 1.243
207 8.2.8.3 11.11
208 9.2.8.3 1.243
209
210 1.3.9.3 9.8824
211 2.3.10.3 9.8824
212 5.3.12.3 9.8824
213 8.3.13.3 9.8824
214
215 /;
216
217
218 parameter phi2(i,j,k,s)
219 /
220
221 1*2.1.1.1 0.02084
222 3.1.1.1 0.02084
223 4*5.1.2.1 0.018518
224 6.1.2.1 0.02084
225 7.1.3.1 0.018518
226 8.1.3.1 0.018518
227 9.1.4.1 0.02084
228
229 1.2.5.1 0.018518
230 3*4.2.6.1 0.02084
231 5.2.7.1 0.02084
232 6.2.11.1 0.26471
233 7.2.7.1 0.02084
234 8*9.2.8.1 0.02084
235
236 1.3.9.1 0.20588
237 2.3.10.1 0.20588
238 5.3.12.1 0.20588
239 8.3.13.1 0.20588
240
241
242 1*2.1.1.2 0.46292
243 3.1.1.2 0.71334
244 4*5.1.2.2 0.6571
245 6.1.2.2 0.5625
246 7.1.3.2 0.71334
```


GAMS 2.25.056 VAX/VMS
simple example of aggregate task synthesis

05/15/93 17:24:28 PAGE 5

```
247      8.1.3.2    0.6571
248      9.1.4.2    0.63545
249
250     1.2.5.2    0.462142
251     3.2.6.2    0.46292
252     4.2.6.2    0.051792
253     5.2.7.2    0.46292
254     6.2.11.2   0.52942
255     7.2.7.2    0.051792
256     8.2.8.2    0.46292
257     9.2.8.2    0.051792
258
259     1.3.9.2    0.41177
260     2.3.10.2   0.41177
261     5.3.12.2   0.41177
262     8.3.13.2   0.41177
263
264     /;
265
266
267 * REACTION 1 IS NITRATION
268 * REACTION 2 IS CHLORINATION
269 * REACTION 3 IS BECHAMP REDUCTION
270 * REACTION 4 IS SANDMEYER REACTION
271
272 * --> MAPPING BETWEEN AGGREGATE TASKS AND CHEMICAL REACTIONS
273 set RMAP(i,j,rxn)
274 /
275 1*2.1.4
276 3.1.1
277 4*5.1.2
278 6.1.4
279 7*8.1.1
280 9.1.2
281
282 1.2.1
283 3*5.2.4
284 6.2.3
285 7*9.2.4
286
287 1*2.3.3
288 5.3.3
289 8.3.3
290 /;
291
292 * --> SIZE OF EQUIPMENT LIST FOR EACH REACTION
293 parameter EQUIPMAP1(rxn)
294 /
295 1 4
296 2 4
297 3 6
298 4 5
299 /;
300
301 * --> SIZE OF EQUIPMENT LIST FOR EACH REACTION
302 parameter EQUIPMAP2(rxn)
303 /
304 1 2
305 2 2
306 /;
307
```

GAMS 2.25.056 VAX/VMS
simple example of aggregate task synthesis

05/15/93 17:24:28 PAGE 6

```
308 * --> SIZE OF UTILITY AND REAGENT LIST FOR EACH REACTION
309 parameter UTILMAP1(rxn)
310 /
311 1 7
312 2 7
313 3 6
314 4 5
315 /;
316
317 * --> SIZE OF UTILITY AND REAGENT LIST FOR EACH REACTION
318 parameter UTILMAP2(rxn)
319 /
320 1 2
321 2 2
322 /;
323
324 * --> EQUIPMENT COSTS MULTIPLYING FACTORS
325 parameter AE1(rxn,1)
326 /
327 1.1 1.1
328 * 1.2 0.18
329 * 1.3 1.4
330 1.4 0.57
331 1.5 0.53
332 1.6 0.12
333 * 1.10 0.1
334
335 2.1 0.85
336 * 2.2 0.17
337 * 2.3 0.08
338 * 2.4 1.4
339 2.5 0.57
340 2.6 0.49
341 2.7 0.1
342 * 2.11 0.11
343
344 3.1 2.4
345 3.2 0.42
346 * 3.3 0.1
347 * 3.4 3.7
348 3.5 0.57
349 3.6 0.46
350 3.7 0.07
351 * 3.8 0.18
352 3.9 0.19
353
354 4.1 1.0
355 * 4.2 0.22
356 4.3 1.2
357 4.4 0.57
358 4.5 0.53
359 4.6 0.09
360 * 4.7 1.3
361 /;
362
363 * --> ADDITIONAL SEPARATIONS
364 parameter AE2(rxn,1)
365 /
366 1.7 0.22
367 1.8 0.05
368 * 1.9 1.5
```

GAMS 2.25.056 VAX/VMS
simple example of aggregate task synthesis

05/15/93 17:24:28 PAGE 7

```
369
370 2.8 0.18
371 2.9 0.05
372 * 2.10 1.35
373 /;
374
375 * --> EQUIPMENT COSTS EXPONENTIATION FACTOR
376 parameter BE1(rxn,1)
377 /
378 1.1 0.481
379 * 1.2 0.519
380 * 1.3 0.624
381 1.4 1.0
382 1.5 0.481
383 1.6 0.519
384 * 1.10 0.519
385
386 2.1 0.481
387 * 2.2 0.519
388 * 2.3 0.519
389 * 2.4 0.624
390 2.5 1.0
391 2.6 0.481
392 2.7 0.519
393 * 2.11 0.519
394
395 3.1 0.481
396 3.2 0.34
397 * 3.3 0.519
398 * 3.4 0.624
399 3.5 1.0
400 3.6 0.481
401 3.7 0.519
402 * 3.8 0.519
403 3.9 0.294
404
405 4.1 0.33
406 * 4.2 0.519
407 4.3 0.33
408 4.4 1.0
409 4.5 0.481
410 4.6 0.519
411 * 4.7 0.73
412 /;
413
414 parameter BE2(rxn,1)
415 /
416 1.7 0.481
417 1.8 0.519
418 * 1.9 0.263
419
420 2.8 0.481
421 2.9 0.519
422 * 2.10 0.263
423 /;
424
425 * --> EQUIPMENT COSTS FIXED TERM
426 parameter CE1(rxn,1)
427 /
428 1.1 0.0
429 * 1.2 0.0
```

GAMS 2.25.056 VAX/VMS
simple example of aggregate task synthesis

05/15/93 17:24:28 PAGE 8

```
430 * 1.3 0.0
431 1.4 2.11
432 1.5 0.0
433 1.6 0.0
434 * 1.10 0.0
435
436 2.1 0.0
437 * 2.2 0.0
438 * 2.3 0.0
439 * 2.4 0.0
440 2.5 2.11
441 2.6 0.0
442 2.7 0.0
443 * 2.11 0.0
444
445 3.1 0.0
446 3.2 0.0
447 * 3.3 0.0
448 * 3.4 0.0
449 3.5 2.11
450 3.6 0.0
451 3.7 0.0
452 * 3.8 0.0
453 3.9 0.0
454
455 4.1 0.0
456 * 4.2 0.0
457 4.3 0.0
458 4.4 2.11
459 4.5 0.0
460 4.6 0.0
461 * 4.7 0.0
462 /;
463
464 parameter CE2(rxn,1)
465 /
466 1.7 0.0
467 1.8 0.0
468 * 1.9 0.0
469
470 2.8 0.0
471 2.9 0.0
472 * 2.10 0.0
473 /;
474
475 * --> UTILITIES AND REAGENTS COSTS MULTIPLYING FACTOR
476 parameter AU1(rxn,1)
477 /
478 1.1 0.18
479 1.2 0.87
480 1.3 6.4
481 1.4 0.45
482 1.7 142.0
483 1.8 1.14
484 1.9 16.25
485
486 2.1 0.24
487 2.2 0.37
488 2.3 5.35
489 2.4 0.36
490 2.7 141.2
```

GAMS 2.25.056 VAX/VMS
simple example of aggregate task synthesis

05/15/93 17:24:28 PAGE 9

```
491 2.8 1.14
492 2.9 16.25
493
494 3.1 5.1
495 3.2 1.03
496 3.3 5.00
497 3.4 25.5
498 3.5 3.68
499 3.6 16.25
500
501 4.1 1.48
502 4.2 10.52
503 4.3 597.0
504 4.4 1.14
505 4.5 16.25
506 /;
507
508 * --> ADDITIONAL SEPARATIONS
509 parameter AU2(rxn,1)
510 /
511 1.5 6.4
512 1.6 0.45
513
514 2.5 0.37
515 2.6 5.25
516 /;
517
518 * --> UTILITIES AND REAGENTS COSTS EXPONENTIATION FACTOR
519 parameter BU1(rxn,1)
520 /
521 1.1 1.0
522 1.2 1.0
523 1.3 1.0
524 1.4 1.0
525 1.7 1.0
526 1.8 1.0
527 1.9 0.25
528
529 2.1 1.0
530 2.2 1.0
531 2.3 1.0
532 2.4 1.0
533 2.7 1.0
534 2.8 1.0
535 2.9 0.25
536
537 3.1 1.0
538 3.2 1.0
539 3.3 1.0
540 3.4 1.0
541 3.5 1.0
542 3.6 0.25
543
544 4.1 1.0
545 4.2 1.0
546 4.3 1.0
547 4.4 1.0
548 4.5 0.25
549 /;
550
551 parameter BU2(rxn,1)
```

GAMS 2.25.056 VAX/VMS
simple example of aggregate task synthesis

05/15/93 17:24:28 PAGE 10

```
552 /
553 1.5 1.0
554 1.6 1.0
555
556 2.5 1.0
557 2.6 1.0
558 /;
559
560 * --> UTILITIES AND REAGENTS COSTS FIXED CHARGE
561 parameter CU1(rxn,1)
562 /
563 1.1 0.0
564 1.2 0.0
565 1.3 0.0
566 1.4 0.0
567 1.7 0.0
568 1.8 0.0
569 1.9 0.0
570
571 2.1 0.0
572 2.2 0.0
573 2.3 0.0
574 2.4 0.0
575 2.7 0.0
576 2.8 0.0
577 2.9 0.0
578
579 3.1 0.0
580 3.2 0.0
581 3.3 0.0
582 3.4 0.0
583 3.5 0.0
584 3.6 0.0
585
586 4.1 0.0
587 4.2 0.0
588 4.3 0.0
589 4.4 0.0
590 4.5 0.0
591 /;
592
593 parameter CU2(rxn,1)
594 /
595 1.5 0.0
596 1.6 0.0
597
598 2.5 0.0
599 2.6 0.0
600 /;
601
602 * --> UTILITIES AND RAW MATERIALS NET PRESENT VALUE FACTOR
603 parameter UCF(rxn,1)
604 /
605 1.1 3.5
606 1.2 3.5
607 1.3 3.5
608 1.4 3.5
609 1.5 3.5
610 1.6 3.5
611 1.7 3.5
612 1.8 3.5
```

GAMS 2.25.056 VAX/VMS
 simple example of aggregate task synthesis

05/15/93 17:24:28 PAGE 11

```

613 1.9 3.5
614
615 2.1 3.5
616 2.2 3.5
617 2.3 3.5
618 2.4 3.5
619 2.5 3.5
620 2.6 3.5
621 2.7 3.5
622 2.8 3.5
623 2.9 3.5
624
625 3.1 3.5
626 3.2 3.5
627 3.3 3.5
628 3.4 3.5
629 3.5 3.5
630 3.6 3.5
631
632 4.1 3.5
633 4.2 3.5
634 4.3 3.5
635 4.4 3.5
636 4.5 3.5
637 /;
638
639 * --> MASS SCALING FACTORS FOR MASS FLOW COSTS
640 * FOR MASS FLOWS WHICH CAN BE PURCHASED, THE CONVERSION FACTOR
641 * IS THE MOLECULAR WEIGHT.
642 * ALL OF THE WASTE STREAMS ARE SCALED AGAINST THE LB-MOLES
643 * DIRECTLY, SO THERE IS NO NEED TO MAKE ANY CORRECTION.
644 parameter MW(k)
645 /
646 1 127.4
647 2 138.0
648 3 127.4
649 4 138.0
650 5 137.4
651 6 172.4
652 7 172.4
653 8 172.4
654 9 182.4
655 10 182.4
656 11 148.0
657 12 182.4
658 13 182.4
659 14 152.4
660 15 152.4
661 16 119.0
662 17 152.4
663 18 152.4
664 19 1.00
665 20 1.00
666 21 1.00
667 22 1.00
668 23 1.00
669 24 1.00
670 25 1.00
671 26 1.00
672 27 1.00
673 28 1.00
  
```

GAMS 2.25.056 VAX/VMS
simple example of aggregate task synthesis

05/15/93 17:24:28 PAGE 12

```
674 29 1.00
675 30 1.00
676 31 1.00
677 32 1.00
678 33 1.00
679 34 1.00
680 35 1.00
681 36 1.00
682 37 1.00
683 38 1.00
684 39 1.00
685 40 1.00
686 /;
687
688 * --> MASS FLOW COST FACTORS
689
690 parameter MFCF(k)
691 /
692 1*40 3.5
693 /;
694
695
696 * --> COST VALUE OF PRODUCT STREAMS (IN $/LB-MOLE FOR FEEDS AND PRODUCTS,
697 * REFERENCED TO LB-MOLE FEED FOR WASTE STREAMS
698 parameter AP(k)
699 /
700 1 2.39
701 2 2.01
702 3 1.85
703 4 2.50
704 5 -5.0
705 6 7.23
706 7 6.85
707 8 8.00
708 9 -5.0
709 10 -5.0
710 11 -5.0
711 12 -5.0
712 13 -5.0
713 14 30.33
714 15 32.67
715 16 19.00
716 17 33.03
717 18 26.95
718 19 -37.76
719 20 -61.00
720 21 -65.00
721 22 -7.1
722 23 -7.1
723 24 -61.0
724 25 -37.76
725 26 -20.0
726 27 -60.5
727 28 -37.76
728 29 -7.1
729 30 -37.76
730 31 -7.1
731 32 -61.0
732 33 -102.0
733 34 -37.76
734 35 -37.76
```


GAMS 2.25.056 VAX/VMS
simple example of aggregate task synthesis

05/15/93 17:24:28 PAGE 13

```
735 36 -7.1
736 37 -20.0
737 38 -185.0
738 39 -128.0
739 40 -156.0
740 /;
741
742 * --> COST VALUE OF FEED STREAMS (IN $/LB-MOLE FOR FEEDS AND PRODUCTS,
743 * REFERENCED TO LB-MOLE FEED FOR WASTE STREAMS
744 parameter AF(k)
745 /
746 1 2.39
747 2 2.01
748 3 1.85
749 4 2.50
750 5 1000.0
751 6 7.23
752 7 6.85
753 8 8.0
754 9 1000.0
755 10 1000.0
756 11 1000.0
757 12 1000.0
758 13 1000.0
759 14 30.33
760 15 32.67
761 16 19.00
762 17 33.03
763 18 26.95
764 19 1000.0
765 20 1000.0
766 21 1000.0
767 22 1000.0
768 23 1000.0
769 24 1000.0
770 25 1000.0
771 26 1000.0
772 27 1000.0
773 28 1000.0
774 29 1000.0
775 30 1000.0
776 31 1000.0
777 32 1000.0
778 33 1000.0
779 34 1000.0
780 35 1000.0
781 36 1000.0
782 37 1000.0
783 38 1000.0
784 39 1000.0
785 40 1000.0
786 /;
787
788 * --> CYCLE TIMES IN HOURS
789 parameter T(i,j)
790 /
791 1.1 24.0
792 1.2 24.0
793 1.3 24.0
794
795 2.1 24.0
```

GAMS 2.25.056 VAX/VMS
simple example of aggregate task synthesis

05/15/93 17:24:28 PAGE 14

```
796 2.3 24.0
797
798 3.1 24.0
799 3.2 24.0
800 3.3 24.0
801
802 4.1 24.0
803 4.2 24.0
804
805 5.1 24.0
806 5.2 24.0
807 5.3 24.0
808
809 6.1 24.0
810 6.2 24.0
811
812 7.1 24.0
813 7.2 24.0
814
815 8.1 24.0
816 8.2 24.0
817 8.3 24.0
818
819 9.1 24.0
820 9.2 24.0
821 /;
822
823 * --> INCREMENTAL CYCLE TIME ASSOCIATED WITH ADDITIONAL SEPARATIONS
824 parameter DT(i,j)
825 /
826 4.1 3.0
827 5.1 3.0
828 7.1 3.0
829 8.1 3.0
830 1.2 3.0
831 2.2 3.0
832 /;
833
834 * --> UPPER BOUND/LOWER BOUND ON BATCH SIZES EXPRESSED AS THE SIZE OF THE
835 * FEED IN POUND MOLES. THE NUMBER IS DERIVED FROM THE SIZE
836 * CONSTRAINT OF 1000 GAL FOR THE FIRST REACTOR IN AN AGGREGATE
837 * TASK.
838
839 parameter FUP(i,j,k)
840 /
841 1*2.1.1 11.11
842 3.1.1 17.12
843 4*5.1.2 15.77
844 6.1.2 13.5
845 7.1.3 17.12
846 8.1.3 15.77
847 9.1.4 15.2507
848
849 1.2.5 11.0914
850 3.2.6 11.11
851 4.2.6 1.243
852 5.2.7 11.11
853 6.2.11 12.7059
854 7.2.7 1.243
855 8.2.8 11.11
856 9.2.8 1.243
```

GAMS 2.25.056 VAX/VMS
simple example of aggregate task synthesis

05/15/93 17:24:28 PAGE 15

```
857
858 1.3.9      9.8824
859 2.3.10    9.8824
860 5.3.12    9.8824
861 8.3.13    9.8824
862
863 /;
864
865
866 parameter FLO(i,j,k)
867 /
868 1*2.1.1    0.5
869 3.1.1      0.5
870 4*5.1.2    0.5
871 6.1.2      0.5
872 7*8.1.3    0.5
873 9.1.4      0.5
874
875 1.2.5      0.5
876 3*4.2.6    0.5
877 5.2.7      0.5
878 6.2.11    6.35295
879 7.2.7      0.5
880 8*9.2.8    0.5
881
882 1.3.9      4.9412
883 2.3.10    4.9412
884 5.3.12    4.9412
885 8.3.13    4.9412
886 /;
887
888 * --> UPPER BOUND ON FEED FLOW RATES SO AS TO ALLOW CHOICE BETWEEN
889 * SEPARATION SEQUENCES. RIGHT NOW IT IS SET TO A "LARGE" VALUE
890 * --> CURRENTLY UNUSED
891 parameter FAUP(i,j,k)
892 /
893 1*3.1.1    2.0
894 4*6.1.2    2.0
895 7*9.1.3    2.0
896
897 1.2.5      2.0
898 3*4.2.6    2.0
899 5.2.7      2.0
900 6.2.11    2.0
901 7.2.7      2.0
902 8*9.2.8    2.0
903
904 1.3.9      2.0
905 2.3.10    2.0
906 5.3.12    2.0
907 8.3.12    2.0
908 /;
909
910 parameter THETA(i,j,k)
911 /
912 * K1
913 1*2.1.5    0.8
914 3.1.6      0.9
915 4.1.6      0.648
916 5.1.7      0.162
917 6.1.11     0.8
```

GAMS 2.25.056 VAX/VMS
simple example of aggregate task synthesis

05/15/93 17:24:28 PAGE 16

```
918 7.1.7 0.63
919 8.1.8 0.27
920 9.1.8 0.81
921
922 1.2.9 0.891
923 3*4.2.10 0.8
924 5.2.12 0.8
925 6.2.16 0.85
926 7.2.12 0.8
927 8*9.2.13 0.8
928
929 1.3.14 0.85
930 2.3.15 0.85
931 5.3.17 0.85
932 8.3.18 0.85
933
934 * K2
935 4.1.7 0.162
936 5.1.6 0.648
937 7.1.8 0.27
938 8.1.7 0.63
939
940 1.2.10 0.009
941 2.2.9 0.891
942
943 * K3
944 4.1.26 1.0
945 5.1.26 1.0
946 7.1.32 1.0
947 8.1.32 1.0
948
949 1.2.20 1.0
950
951 * K4
952 1*2.1.19 1.0
953 3.1.24 1.0
954 4.1.27 1.0
955 5.1.39 1.0
956 6.1.30 1.0
957 7.1.33 1.0
958 8.1.40 1.0
959 9.1.37 1.0
960
961 1.2.21 1.0
962 3*4.2.25 1.0
963 5.2.28 1.0
964 6.2.31 1.0
965 7.2.34 1.0
966 8*9.2.35 1.0
967
968 1.3.22 1.0
969 2.3.23 1.0
970 5.3.29 1.0
971 8.3.36 1.0
972 /;
973
974 scalar NP1MAX /0.35/ ;
975 scalar NP2MAX /0.35/ ;
976 scalar NP3MAX /0.45/ ;
977 scalar NP4MAX /0.35/ ;
978 scalar NP5MAX /0.35/ ;
```

GAMS 2.25.056 VAX/VMS
simple example of aggregate task synthesis

05/15/93 17:24:28 PAGE 17

```
979
980 positive variables
981     pa(i,j,k)      ,
982     fa(i,j,k)      ,
983     nfa(k)         ,
984     npa(k)         ,
985     f(i,j,k)       ,
986     equip1(i,j)    ,
987     util1(i,j)     ,
988     equip2(i,j)    ,
989     util2(i,j)     ,
990     lam(i,j,k,q)   ,
991     lam2(i,j,k,s)  ,
992     lam3(i,j,k,q)  ,
993     lam4(i,j,k,q)  ,
994     lam5(i,j,k,s)  ,
995     lam6(i,j,k,s)  ;
996
997 binary variables
998     y(i,j)         ,
999     x(i,j)         ,
1000    del(i,j,k,q)   ,
1001    del2(i,j,k,s)  ;
1002
1003 variables
1004     z               ;
1005
1006 equations
1007     abal(k)        ,
1008     fdef1(i,j,k)   ,
1009     exst1(i,j)     ,
1010     exst2(i,j)     ,
1011     ibal0(i,j,k,k) ,
1012     ibal2(i,j,k,k) ,
1013     ibal3(i,j,k,k) ,
1014     ibal4(i,j)     ,
1015     fset(i,j,k)    ,
1016     faset(i,j,k)   ,
1017     pick1(i,j,k,q1,q2) ,
1018     pick2(i,j,k)   ,
1019     pick3(i,j,k,s1,s2) ,
1020     pick4(i,j,k)   ,
1021     pick5(i,j,k)   ,
1022     pick6(i,j,k)   ,
1023     pick7(i,j,k,q) ,
1024     pick9(i,j,k,s) ,
1025     ecost1(i,j,k,rxn) ,
1026     ucost1(i,j,k,rxn) ,
1027     ucost2a(i,j,k,rxn) ,
1028     ecost2b(i,j,k,rxn) ,
1029     ecost2c(i,j,k,rxn) ,
1030     ucost2c(i,j,k,rxn) ,
1031     xupb(i,j)      ,
1032     yupb(i,j)      ,
1033     pcon4          ,
1034     ldef1(i,j,k,q) ,
1035     ldef2(i,j,k,q) ,
1036     ldef3(i,j,k,q) ,
1037     ldef4(i,j,k,q) ,
1038     ldef5(i,j,k,q) ,
1039     ldef6(i,j,k,q) ,
```

simple example of aggregate task synthesis

```

1040     ldef7(i,j,k,s)      ,
1041     ldef8(i,j,k,s)      ,
1042     ldef9(i,j,k,s)      ,
1043     ldef10(i,j,k,s)     ,
1044     ldef11(i,j,k,s)     ,
1045     ldef12(i,j,k,s)     ,
1046     tc1,tc2,tc3,tc4,tc5,
1047     sym1,sym2,sym3,sym5,
1048     sym6,sym7,
1049     obj                    ;
1050
1051 * AGGREGATE TASK MASS BALANCES
1052 abal(k).. sum((i,j)$K1(i,j,k),pa(i,j,k)) + sum((i,j)$K2(i,j,k),pa(i,j,k)
1053                                     ) +
1054                                     sum((i,j)$K3(i,j,k),pa(i,j,k)) + sum((i,j)$K4(i,j,k),pa(i,j,k)
1055                                     )-
1056                                     npa(k)$PRDD(k)
1057                                     =e= sum((i,j)$K0(i,j,k),fa(i,j,k)) -
1058                                     nfa(k)$FEED(k);
1059
1060 * DEFINITION OF TIME-AVERAGED MASS FLOWS
1061 fdef1(i,j,k)$IJ(i,j) $K0(i,j,k)..
1062     fa(i,j,k) =e= f(i,j,k)/T(i,j) - sum(q,lam3(i,j,k,q)*phi(i,j,k,q))
1063                                     $SEPX(i,j) *
1064                                     DT(i,j)$SEPX(i,j)/(T(i,j)*(T(i,j)+DT(i,j)$SEPX(i,j)));
1065
1066 * EXISTENCE OF AGGREGATE TASKS
1067 exst1(i,j)$IJ(i,j).. sum(k$K0(i,j,k),f(i,j,k)) =l=
1068     sum(k$K0(i,j,k),FUP(i,j,k))*y(i,j);
1069 exst2(i,j)$IJ(i,j).. sum(k$K0(i,j,k),f(i,j,k)) =g=
1070     sum(k$K0(i,j,k),FLO(i,j,k))*y(i,j);
1071
1072 * INTERNAL MASS BALANCES
1073 ibal0(i,j,k,m)$IJ(i,j)$K0(i,j,k)$K1(i,j,m)..
1074     pa(i,j,m) =e= THETA(i,j,m)*fa(i,j,k);
1075
1076 ibal2(i,j,k,m)$IJ(i,j)$K0(i,j,k)$K2(i,j,m)+K3(i,j,m)..
1077     pa(i,j,m) =l= THETA(i,j,m)*
1078     sum(s,lam5(i,j,k,s)*phi2(i,j,k,s))$SEPX(i,j);
1079
1080 ibal3(i,j,k,m)$IJ(i,j)$K0(i,j,k)$K2(i,j,m)+K3(i,j,m)..
1081     pa(i,j,m) =g= THETA(i,j,m)*fa(i,j,k) -
1082     THETA(i,j,m)*
1083     sum(s,lam6(i,j,k,s)*phi2(i,j,k,s))$SEPX(i,j);
1084
1085 ibal4(i,j)$IJ(i,j)..
1086     0.5*(sum(k$K2(i,j,k),pa(i,j,k)/THETA(i,j,k)) +
1087     sum(k$K3(i,j,k),pa(i,j,k)/THETA(i,j,k)) ) +
1088     sum(k$K4(i,j,k),pa(i,j,k)/THETA(i,j,k))
1089     =e= sum(k$K0(i,j,k),fa(i,j,k));
1090
1091 * EQUATIONS FOR DISCRETIZING FLOWS
1092 fset(i,j,k)$IJ(i,j) $K0(i,j,k)..
1093     f(i,j,k) =e= sum(q,lam(i,j,k,q)*phi(i,j,k,q));
1094
1095 faset(i,j,k)$IJ(i,j) $K0(i,j,k)..
1096     fa(i,j,k) =e= sum(s,lam2(i,j,k,s)*phi2(i,j,k,s));
1097

```

GAMS 2.25.056 VAX/VMS 05/15/93 17:24:28 PAGE 19
 simple example of aggregate task synthesis

```

1098 pick1(i,j,k,q1,q2)$ ( IJ(i,j) $K0(i,j,k) $(ord(q2) eq (ord(q1)-1)) )..
1099     lam(i,j,k,q1) + lam(i,j,k,q2) =g= del(i,j,k,q2);
1100
1101 pick2(i,j,k)$ ( IJ(i,j) $K0(i,j,k) )..
1102     sum(q$(ord(q) lt 3), del(i,j,k,q)) =e= y(i,j);
1103
1104 pick5(i,j,k)$ ( IJ(i,j) $K0(i,j,k) )..
1105     sum(q, lam(i,j,k,q)) =l= y(i,j);
1106
1107 pick7(i,j,k,q2)$ ( IJ(i,j) $K0(i,j,k) $(ord(q2) lt 3))..
1108     del(i,j,k,q2) +
1109     sum(q1$(ord(q1) gt (ord(q2)+1)), lam(i,j,k,q1)) +
1110     sum(q1$(ord(q1) lt ord(q2)), lam(i,j,k,q1)) =l= y(i,j);
1111
1112 pick3(i,j,k,s1,s2)$ ( IJ(i,j) $K0(i,j,k) $(ord(s2) eq (ord(s1)-1)) )..
1113     lam2(i,j,k,s1) + lam2(i,j,k,s2) =g= del2(i,j,k,s2);
1114
1115 pick4(i,j,k)$ ( IJ(i,j) $K0(i,j,k) )..
1116     sum(s$(ord(s) lt 2), del2(i,j,k,s)) =e= y(i,j);
1117
1118 pick6(i,j,k)$ ( IJ(i,j) $K0(i,j,k) )..
1119     sum(s, lam2(i,j,k,s)) =l= y(i,j);
1120
1121 pick9(i,j,k,s2)$ ( IJ(i,j) $K0(i,j,k) $(ord(s2) lt 2))..
1122     del2(i,j,k,s2) +
1123     sum(s1$(ord(s1) gt (ord(s2)+1)), lam2(i,j,k,s1)) +
1124     sum(s1$(ord(s1) lt ord(s2)), lam2(i,j,k,s1)) =l= y(i,j);
1125
1126 * EQUIPMENT COST FOR EACH AGGREGATE TASK (ADDITIONAL SEPARATIONS NOT
                                     INCLUDED)
1127
1128 ecost1(i,j,k,rxn)$ (RMAP(i,j,rxn) $K0(i,j,k))..
1129     equip1(i,j) =g=
1130     sum(l$AE1(rxn,l), AE1(rxn,l)*sum(q, lam(i,j,k,q)*phi(i,j,k,q)
                                     **BE1(rxn,l))
1131     + CE1(rxn,l)*y(i,j));
1132
1133 * UTILITY AND REAGENT COSTS FOR EACH AGGREGATE TASK (ADDITIONAL
                                     SEPARATIONS NOT
1134 * INCLUDED)
1135 ucost1(i,j,k,rxn)$ (RMAP(i,j,rxn) $K0(i,j,k))..
1136     util1(i,j) =g=
1137     sum(l$AU1(rxn,l), UCF(rxn,l)*(AU1(rxn,l)*sum(s, lam2(i,j,k,s)*
1138     phi2(i,j,k,s)**BU1(rxn,l))
1139     + CU1(rxn,l)*y(i,j)));
1140
1141 * ADDITIONAL SEPARATION EQUIPMENT COST FOR EACH AGGREGATE TASK
1142 ecost2b(i,j,k,rxn)$ (RMAP(i,j,rxn) $K0(i,j,k))..
1143     equip2(i,j) =l= sum(l$AE2(rxn,l),
1144     AE2(rxn,l)*sum(q, lam3(i,j,k,q)*phi(i,j,k,q)**BE2(rxn,l))
1145     + CE2(rxn,l)*x(i,j));
1146
1147 ccost2c(i,j,k,rxn)$ (RMAP(i,j,rxn) $K0(i,j,k) $SEPX(i,j))..
1148     equip2(i,j) =g=
1149     sum(l$AE2(rxn,l),
1150     AE2(rxn,l)*sum(q, lam(i,j,k,q)*phi(i,j,k,q)**BE2(rxn,l))
1151     + CE2(rxn,l)*x(i,j))
1152     -
1153     sum(l$AE2(rxn,l),
1154     AE2(rxn,l)*sum(q, lam4(i,j,k,q)*phi(i,j,k,q)**BE2(rxn,l))
1155     + CE2(rxn,l)*x(i,j));

```

GAMS 2.25.056 VAX/VMS
simple example of aggregate task synthesis

05/15/93 17:24:28 PAGE 20

```

1156
1157 ldef1(i,j,k,q)$(K0(i,j,k) $SEPX(i,j))..
1158     lam3(i,j,k,q) =l= x(i,j);
1159 ldef2(i,j,k,q)$(K0(i,j,k) $SEPX(i,j))..
1160     lam3(i,j,k,q) =l= lam(i,j,k,q);
1161 ldef3(i,j,k,q)$(K0(i,j,k) $SEPX(i,j))..
1162     lam3(i,j,k,q) =g= lam(i,j,k,q) + x(i,j) - 1;
1163
1164 ldef4(i,j,k,q)$(K0(i,j,k) $SEPX(i,j))..
1165     lam4(i,j,k,q) =l= 1-x(i,j);
1166 ldef5(i,j,k,q)$(K0(i,j,k) $SEPX(i,j))..
1167     lam4(i,j,k,q) =l= lam(i,j,k,q);
1168 ldef6(i,j,k,q)$(K0(i,j,k) $SEPX(i,j))..
1169     lam4(i,j,k,q) =g= lam(i,j,k,q) - x(i,j);
1170
1171
1172 * ADDITIONAL SEPARATION UTILITY COSTS
1173 .cost2a(i,j,k,rxn)$(RMAP(i,j,rxn) $K0(i,j,k) $SEPX(i,j))..
1174     util2(i,j) =l=
1175     sum(1$AU2(rxn,l),
1176     UCF(rxn,l)*(AU2(rxn,l)*sum(s,lam5(i,j,k,s)*phi2(i,j,k,s)**BU2(rxn,l))
1177     + CU2(rxn,l)*y(i,j)));
1178
1179 ucost2c(i,j,k,rxn)$(RMAP(i,j,rxn) $K0(i,j,k) $SEPX(i,j))..
1180     util2(i,j) =g=
1181     sum(1$AU2(rxn,l),
1182     UCF(rxn,l)*(AU2(rxn,l)*sum(s,lam2(i,j,k,s)*phi2(i,j,k,s)**BU2(rxn,l))
1183     + CU2(rxn,l)*y(i,j)))
1184     -
1185     sum(1$AU2(rxn,l),
1186     UCF(rxn,l)*(AU2(rxn,l)*sum(s,lam6(i,j,k,s)*phi2(i,j,k,s)**BU2(rxn,l))
1187     + CU2(rxn,l)*y(i,j)));
1188
1189 ldef7(i,j,k,s)$(K0(i,j,k) $SEPX(i,j))..
1190     lam5(i,j,k,s) =l= x(i,j);
1191 ldef8(i,j,k,s)$(K0(i,j,k) $SEPX(i,j))..
1192     lam5(i,j,k,s) =l= lam2(i,j,k,s);
1193 ldef9(i,j,k,s)$(K0(i,j,k) $SEPX(i,j))..
1194     lam5(i,j,k,s) =g= lam2(i,j,k,s) + x(i,j) - 1;
1195
1196 ldef10(i,j,k,s)$(K0(i,j,k) $SEPX(i,j))..
1197     lam6(i,j,k,s) =l= 1-x(i,j);
1198 ldef11(i,j,k,s)$(K0(i,j,k) $SEPX(i,j))..
1199     lam6(i,j,k,s) =l= lam2(i,j,k,s);
1200 ldef12(i,j,k,s)$(K0(i,j,k) $SEPX(i,j))..
1201     lam6(i,j,k,s) =g= lam2(i,j,k,s) - x(i,j);
1202
1203
1204 * UPPER BOUNDS FOR BINARY VARIABLES DURING RELAXED SOLUTION
1205 yupb(i,j)$IJJ(i,j).. y(i,j) =l= 1;
1206 xupb(i,j)$SEPX(i,j).. x(i,j) =l= y(i,j);
1207
1208 * PRODUCTION CONSTRAINT
1209 pcon4.. y('1','3')+y('2','3')+y('5','3')+y('8','3')+y('6','2') =g= 4;
1210
1211
1212 * TOPOLOGICAL CONSTRAINTS (SEE SYMMETRY BREAKING CONSTRAINTS BELOW)
1213 tc1.. y('1','3') =l= y('1','2');
1214 tc2.. y('1','2') =l= y('1','1');
1215 tc3.. y('2','3') =l= y('1','2')+y('3','2');
1216 tc4.. y('5','3') =l= y('5','2');

```


GAMS 2.25.056 VAX/VMS 05/15/93 17:24:28 PAGE 21
 simple example of aggregate task synthesis

```

1217 tc5.. y('8','3') =1= y('8','2');
1218
1219 * SYMMETRY BREAKING CONSTRAINTS
1220 sym1.. y('2','1') =1= y('1','1');
1221 sym2.. y('5','1') =1= y('4','1') + 2 - x('4','1') - x('5','1');
1222 sym3.. y('8','1') =1= y('7','1') + 2 - x('7','1') - x('8','1');
1223 sym5.. y('4','2') =1= y('3','2');
1224 sym6.. y('7','2') =1= y('5','2');
1225 sym7.. y('9','2') =1= y('8','2');
1226
1227 * OBJECTIVE FUNCTION (M&S RATIO = 1.7)
1228
1229 obj.. z =e=
1230     0.7*sum(k$PRDD2(k),MFCF(k)*MW(k)*AP(k)*npa(k))
1231     -
1232     7.6*1.7*(sum((i,j)$IJ(i,j),equip1(i,j)) +
1233             sum((i,j)$SEPX(i,j),equip2(i,j))) -
1234             (sum((i,j)$IJ(i,j),util1(i,j)) +
1235              sum((i,j)$SEPX(i,j),util2(i,j)) +
1236              sum(k$FEED(k),MFCF(k)*MW(k)*AF(k)*nfa(k)) -
1237              sum(k$WASTE(k),MFCF(k)*MW(k)*AP(k)*npa(k))
1238             );
1239
1240
1241 * UPPER BOUNDS // PRODUCTION QUOTAS
1242 npa.up('14') = NP1MAX;
1243 npa.up('15') = NP2MAX;
1244 npa.up('16') = NP3MAX;
1245 npa.up('17') = NP4MAX;
1246 npa.up('18') = NP5MAX;
1247
1248 lam.up(i,j,k,q)$K0(i,j,k) =1;
1249 lam2.up(i,j,k,s)$K0(i,j,k) =1;
1250 lam3.up(i,j,k,q)$K0(i,j,k)$SEPX(i,j) =1;
1251 lam4.up(i,j,k,q)$K0(i,j,k)$SEPX(i,j) =1;
1252 lam5.up(i,j,k,s)$K0(i,j,k)$SEPX(i,j) =1;
1253 lam6.up(i,j,k,s)$K0(i,j,k)$SEPX(i,j) =1;
1254
1255 option limrow = 0;
1256 option limcol = 0;
1257 option iterlim = 200000;
1258 option reslim = 200000;
1259 option optca =0.0;
1260 option optcr =0.0;
1261 model aggtask /all/;
1262 aggtask.optfile=1;
1263
1264 solve aggtask using mip maximizing z;

```

COMPILATION TIME = 1.940 SECONDS VERID VMS-00-056

GAMS 2.25.056 VAX/VMS 05/15/93 17:24:28 PAGE 22
simple example of aggregate task synthesis
Model Statistics SOLVE AGGTASK USING MIP FROM LINE 1264

MODEL STATISTICS

BLOCKS OF EQUATIONS	51	SINGLE EQUATIONS	684
BLOCKS OF VARIABLES	20	SINGLE VARIABLES	493
NON ZERO ELEMENTS	2039	DISCRETE VARIABLES	89

GENERATION TIME = 3.910 SECONDS

EXECUTION TIME = 4.160 SECONDS VERID VMS-00-056

GAMS 2.25.056 VAX/VMS 05/15/93 17:24:28 PAGE 23
 simple example of aggregate task synthesis
 Solution Report SOLVE AGGTASK USING MIP FROM LINE 1264

S O L V E S U M M A R Y

MODEL AGGTASK OBJECTIVE Z
 TYPE MIP DIRECTION MAXIMIZE
 SOLVER OSL FROM LINE 1264

**** SOLVER STATUS 1 NORMAL COMPLETION
 **** MODEL STATUS 1 OPTIMAL
 **** OBJECTIVE VALUE 4513.1188

RESOURCE USAGE, LIMIT 69.840 200000.000
 ITERATION COUNT, LIMIT 592 200000

--- GAMS/OSL 2.001

Could not find option file SYSS\$USER2:[D232JK60]OSL.OPT
 Using defaults instead.

Work space allocated -- 0.77 Mb

OSL step: Reading data 0.28 Seconds

OSL step: Scale 0.42 Seconds
 Range of matrix coefficients:
 before scaling : 0.0002 - 638400.0000
 after scaling : 0.0091 - 1.0000

OSL step: Presolve 0.34 Seconds
 Size reduction:
 rows : 684 (old) 595 (new)
 columns : 493 (old) 493 (new)
 nonzeros: 2039 (old) 1895 (new)

OSL step: Crash 0.22 Seconds
 Crash option : 1

OSL step: Primal Simplex 6.28 Seconds
 Status: Successful (optimal)
 Iterations: 260
 Objective : 4627.3536

OSL step: Branch&Bound 60.83 Seconds
 Status: Successful (optimal)
 Iterations: 332
 Nodes : 119
 Objective : 4513.1188

OSL step: Postsolve 0.35 Seconds

OSL step: Primal Simplex 0.37 Seconds
 Status: Successful (optimal)
 Iterations: 0
 Objective : 4513.1188

GAMS 2.25.056 VAX/VMS

05/15/93 17:24:28 PAGE 24

simple example of aggregate task synthesis

Solution Report SOLVE AGGTASK USING MIP FROM LINE 1264

---- EQU ABAL

	LOWER	LEVEL	UPPER	MARGINAL
1	.	.	.	-1065.701
2	.	.	.	-970.830
3	.	.	.	-824.915
4	.	.	.	-1207.500
5	.	.	.	-4773.408
6	.	.	.	-3053.807
7	.	.	.	-2893.303
8	.	.	.	-3379.040
9	.	.	.	-6582.920
10	.	.	.	-7499.250
11	.	.	.	-4511.278
12	.	.	.	-7298.619
13	.	.	.	-8014.970
14	.	.	.	-8377.827
15	.	.	.	-9455.861
16	.	.	.	-5539.450
17	.	.	.	-9219.825
18	.	.	.	-1.006E+4
19	.	.	.	132.160
20	.	.	.	213.500
21	.	.	.	227.500
22	.	.	.	24.850
23	.	.	.	24.850
24	.	.	.	213.500
25	.	.	.	132.160
26	.	.	.	70.000
27	.	.	.	211.750
28	.	.	.	132.160
29	.	.	.	24.850
30	.	.	.	132.160
31	.	.	.	24.850
32	.	.	.	213.500
33	.	.	.	357.000
34	.	.	.	132.160
35	.	.	.	132.160
36	.	.	.	24.850
37	.	.	.	70.000
38	.	.	.	-3500.000
39	.	.	.	448.000
40	.	.	.	546.000

---- EQU FDEF1

	LOWER	LEVEL	UPPER	MARGINAL
1.1.1	.	.	.	428.137
1.2.5	.	.	.	277.422
1.3.9	.	.	.	337.089
2.1.1	.	.	.	428.137
2.3.10	.	.	.	337.089
3.1.1	.	.	.	896.312
3.2.6	.	.	.	620.704
4.1.2	.	.	.	819.439
4.2.6	.	.	.	499.512
5.1.2	.	.	.	819.439
5.2.7	.	.	.	620.704

GAMS 2.25.056 VAX/VMS 05/15/93 17:24:28 PAGE 25
 simple example of aggregate task synthesis
 Solution Report SOLVE AGGTASK USING MIP FROM LINE 1264

EQU FDEF1

	LOWER	LEVEL	UPPER	MARGINAL
5.3.12	.	.	.	337.089
6.1.2	.	.	.	282.978
6.2.11	.	.	.	414.419
7.1.3	.	.	.	1099.061
7.2.7	.	.	.	499.512
8.1.3	.	.	.	1096.758
8.2.8	.	.	.	708.048
8.3.13	.	.	.	337.089
9.1.4	.	.	.	891.751
9.2.8	.	.	.	813.717

---- EQU EXST1

	LOWER	LEVEL	UPPER	MARGINAL
1.1	-INF	.	.	.
1.2	-INF	-9.479E-5	.	.
1.3	-INF	-4.706E-5	.	.
2.1	-INF	-8.356	.	.
2.3	-INF	-4.706E-5	.	.
3.1	-INF	.	.	.
3.2	-INF	.	.	.
4.1	-INF	.	.	17.778
4.2	-INF	-5.882E-5	.	.
5.1	-INF	.	.	17.778
5.2	-INF	.	.	.
5.3	-INF	-4.706E-5	.	.
6.1	-INF	.	.	.
6.2	-INF	.	.	.
7.1	-INF	.	.	27.308
7.2	-INF	-5.882E-5	.	.
8.1	-INF	.	.	26.962
8.2	-INF	.	.	.
8.3	-INF	-0.994	.	.
9.1	-INF	.	.	.
9.2	-INF	.	.	.

---- EQU EXST2

	LOWER	LEVEL	UPPER	MARGINAL
1.1	.	10.610	+INF	.
1.2	.	10.591	+INF	.
1.3	.	4.941	+INF	.
2.1	.	2.254	+INF	.
2.3	.	4.941	+INF	.
3.1	.	16.620	+INF	.
3.2	.	10.610	+INF	.
4.1	.	15.270	+INF	.
4.2	.	0.743	+INF	.
5.1	.	15.270	+INF	.
5.2	.	10.610	+INF	.
5.3	.	4.941	+INF	.
6.1	.	.	+INF	-2.650
6.2	.	.	+INF	.

GAMS 2.25.056 VAX/VMS

05/15/93 17:24:28 PAGE 26

simple example of aggregate task synthesis

Solution Report SOLVE AGGTASK USING MIF FROM LINE 1264

EQU EXST2

	LOWER	LEVEL	UPPER	MARGINAL
7.1	.	16.620	+INF	.
7.2	.	0.743	+INF	.
8.1	.	15.270	+INF	.
8.2	.	10.610	+INF	.
8.3	.	3.947	+INF	.
9.1	.	14.751	+INF	.
9.2	.	.	+INF	.

---- EQU IBAL0

	LOWER	LEVEL	UPPER	MARGINAL
1.1.1 .5	.	.	.	4773.408
1.2.5 .9	.	.	.	6582.920
1.3.9 .14	.	.	.	8377.827
2.1.1 .5	.	.	.	4773.408
2.3.10.15	.	.	.	9455.861
3.1.1 .6	.	.	.	3053.807
3.2.6 .10	.	.	.	7499.250
4.1.2 .6	.	.	.	3053.807
4.2.6 .10	.	.	.	7499.250
5.1.2 .7	.	.	.	2893.303
5.2.7 .12	.	.	.	7298.619
5.3.12.17	.	.	.	9219.825
6.1.2 .11	.	.	.	4511.278
6.2.11.16	.	.	.	5539.450
7.1.3 .7	.	.	.	2893.303
7.2.7 .12	.	.	.	7298.619
8.1.3 .8	.	.	.	3379.040
8.2.8 .13	.	.	.	8014.970
8.3.13.18	.	.	.	10062.591
9.1.4 .8	.	.	.	3379.040
9.2.8 .13	.	.	.	8014.970

---- EQU IBAL2

	LOWER	LEVEL	UPPER	MARGINAL
1.2.5.10	-INF	.	.	20138.139
1.2.5.20	-INF	.	.	.
4.1.2.7	-INF	.	.	3546.852
4.1.2.26	-INF	.	.	35.875
5.1.2.6	-INF	.	.	3399.486
5.1.2.26	-INF	.	.	154.000
7.1.3.8	-INF	.	.	4040.151
7.1.3.32	-INF	.	.	.
8.1.3.7	-INF	.	.	3326.636
8.1.3.32	-INF	.	.	59.500

GAMS 2.25.056 VAX/VMS 05/15/93 17:24:28 PAGE 27
 simple example of aggregate task synthesis
 Solution Report SOLVE AGGTASK USING MIP FROM LINE 1264

---- EQU IBAL3

	LOWER	LEVEL	UPPER	MARGINAL
1.2.5.10	.	.	+INF	.
1.2.5.20	.	.	+INF	.
4.1.2.7	.	.	+INF	.
4.1.2.26	.	.	+INF	.
5.1.2.6	.	.	+INF	.
5.1.2.26	.	.	+INF	.
7.1.3.8	.	.	+INF	.
7.1.3.32	.	.	+INF	-35.000
8.1.3.7	.	.	+INF	.
8.1.3.32	.	.	+INF	.

---- EQU IBAL4

	LOWER	LEVEL	UPPER	MARGINAL
1.1	.	.	.	-132.160
1.2	.	.	.	-227.500
1.3	.	.	.	-24.850
2.1	.	.	.	-132.160
2.3	.	.	.	-24.850
3.1	.	.	.	-213.500
3.2	.	.	.	-132.160
4.1	.	.	.	-211.750
4.2	.	.	.	-132.160
5.1	.	.	.	-448.000
5.2	.	.	.	-132.160
5.3	.	.	.	-24.850
6.1	.	.	.	-132.160
6.2	.	.	.	-24.850
7.1	.	.	.	-357.000
7.2	.	.	.	-132.160
8.1	.	.	.	-546.000
8.2	.	.	.	-132.160
8.3	.	.	.	-24.850
9.1	.	.	.	-70.000
9.2	.	.	.	-132.160

---- EQU FSET

	LOWER	LEVEL	UPPER	MARGINAL
1.1.1	.	.	.	17.839
1.2.5	.	.	.	11.559
1.3.9	.	.	.	14.045
2.1.1	.	.	.	17.839
2.3.10	.	.	.	14.045
3.1.1	.	.	.	37.346
3.2.6	.	.	.	25.863
4.1.2	.	.	.	16.365
4.2.6	.	.	.	20.813
5.1.2	.	.	.	16.365
5.2.7	.	.	.	25.863
5.3.12	.	.	.	14.045
6.1.2	.	.	.	14.441
6.2.11	.	.	.	17.267

GAMS 2.25.056 VAX/VMS

05/15/93 17:24:28 PAGE 28

simple example of aggregate task synthesis

Solution Report SOLVE AGGTASK USING MIP FROM LINE 1264

EQU FSET

	LOWER	LEVEL	UPPER	MARGINAL
7.1.3	.	.	.	18.486
7.2.7	.	.	.	20.813
8.1.3	.	.	.	18.736
8.2.8	.	.	.	29.502
8.3.13	.	.	.	14.045
9.1.4	.	.	.	37.156
9.2.8	.	.	.	33.905

---- EQU FASET

	LOWER	LEVEL	UPPER	MARGINAL
1.1.1	.	.	.	2192.728
1.2.5	.	.	.	587.052
1.3.9	.	.	.	176.293
2.1.1	.	.	.	2192.728
2.3.10	.	.	.	176.293
3.1.1	.	.	.	572.914
3.2.6	.	.	.	2192.728
4.1.2	.	.	.	-23.152
4.2.6	.	.	.	2313.920
5.1.2	.	.	.	-1769.554
5.2.7	.	.	.	2192.728
5.3.12	.	.	.	176.293
6.1.2	.	.	.	2223.055
6.2.11	.	.	.	-242.015
7.1.3	.	.	.	-493.195
7.2.7	.	.	.	2313.920
8.1.3	.	.	.	-1555.332
8.2.8	.	.	.	2192.728
8.3.13	.	.	.	176.293
9.1.4	.	.	.	567.772
9.2.8	.	.	.	2087.059

---- EQU PICK1

	LOWER	LEVEL	UPPER	MARGINAL
1.1.1 .2.1	.	.	+INF	-11.546
1.1.1 .3.2	.	.	+INF	-11.546
1.2.5 .2.1	.	1.1715E-5	+INF	.
1.2.5 .3.2	.	.	+INF	-25.880
1.3.9 .2.1	.	1.6326E-5	+INF	.
1.3.9 .3.2	.	.	+INF	-64.797
2.1.1 .2.1	.	.	+INF	-23.093
2.1.1 .3.2	.	0.902	+INF	.
2.3.10.2.1	.	1.6326E-5	+INF	.
2.3.10.3.2	.	.	+INF	-64.797
3.1.1 .2.1	.	.	+INF	.
3.1.1 .3.2	.	.	+INF	.
3.2.6 .2.1	.	.	+INF	.
3.2.6 .3.2	.	.	+INF	.
4.1.2 .2.1	.	.	+INF	-4.650
4.1.2 .3.2	.	.	+INF	.
4.2.6 .2.1	.	1.1932E-4	+INF	.

GAMS 2.25.056 VAX/VMS 05/15/93 17:24:28 PAGE 29
 simple example of aggregate task synthesis
 Solution Report SOLVE AGGTASK USING MIP FROM LINE 1264

EQU PICK1

	LOWER	LEVEL	UPPER	MARGINAL
4.2.6 .3.2	.	.	+INF	-22.727
5.1.2 .2.1	.	.	+INF	.
5.1.2 .3.2	.	.	+INF	.
5.2.7 .2.1	.	.	+INF	.
5.2.7 .3.2	.	.	+INF	.
5.3.12.2.1	.	1.6326E-5	+INF	.
5.3.12.3.2	.	.	+INF	-64.797
6.1.2 .2.1	.	.	+INF	.
6.1.2 .3.2	.	.	+INF	.
6.2.11.2.1	.	.	+INF	-21.121
6.2.11.3.2	.	.	+INF	-21.121
7.1.3 .2.1	.	.	+INF	.
7.1.3 .3.2	.	.	+INF	.
7.2.7 .2.1	.	1.1932E-4	+INF	.
7.2.7 .3.2	.	.	+INF	-22.727
8.1.3 .2.1	.	.	+INF	.
8.1.3 .3.2	.	.	+INF	.
8.2.8 .2.1	.	.	+INF	.
8.2.8 .3.2	.	.	+INF	.
8.3.13.2.1	.	0.345	+INF	.
8.3.13.3.2	.	.	+INF	-64.797
9.1.4 .2.1	.	.	+INF	.
9.1.4 .3.2	.	.	+INF	.
9.2.8 .2.1	.	.	+INF	-6.454
9.2.8 .3.2	.	.	+INF	-6.454

---- EQU PICK2

	LOWER	LEVEL	UPPER	MARGINAL
1.1.1	.	.	.	-11.546
1.2.5	.	.	.	-25.880
1.3.9	.	.	.	EPS
2.1.1	.	.	.	-50.688
2.3.10	.	.	.	EPS
3.1.1	.	.	.	EPS
3.2.6	.	.	.	EPS
4.1.2	.	.	.	EPS
4.2.6	.	.	.	-22.727
5.1.2	.	.	.	EPS
5.2.7	.	.	.	EPS
5.3.12	.	.	.	EPS
6.1.2	.	.	.	EPS
6.2.11	.	.	.	-21.121
7.1.3	.	.	.	EPS
7.2.7	.	.	.	-22.727
8.1.3	.	.	.	EPS
8.2.8	.	.	.	EPS
8.3.13	.	.	.	EPS
9.1.4	.	.	.	EPS
9.2.8	.	.	.	-6.454

GAMS 2.25.056 VAX/VMS

05/15/93 17:24:28 PAGE 30

simple example of aggregate task synthesis

Solution Report SOLVE AGGTASK USING MIP FROM LJNE 1264

---- EQU PICK3

	LOWER	LEVEL	UPPER	MARGINAL
1.1.1 .2.1	.	.	+INF	-20.417
1.2.5 .2.1	.	.	+INF	-19.899
1.3.9 .2.1	.	.	+INF	-31.062
2.1.1 .2.1	.	.	+INF	-20.417
2.3.10.2.1	.	.	+INF	-31.062
3.1.1 .2.1	.	.	+INF	-20.687
3.2.6 .2.1	.	.	+INF	-20.417
4.1.2 .2.1	.	.	+INF	-20.104
4.2.6 .2.1	.	.	+INF	-17.891
5.1.2 .2.1	.	.	+INF	-20.104
5.2.7 .2.1	.	.	+INF	-20.417
5.3.12.2.1	.	.	+INF	-31.062
6.1.2 .2.1	.	.	+INF	.
6.2.11.2.1	.	.	+INF	-142.206
7.1.3 .2.1	.	.	+INF	-20.147
7.2.7 .2.1	.	.	+INF	-17.891
8.1.3 .2.1	.	.	+INF	-20.104
8.2.8 .2.1	.	.	+INF	-20.417
8.3.13.2.1	.	.	+INF	-31.062
9.1.4 .2.1	.	.	+INF	-20.620
9.2.8 .2.1	.	.	+INF	-22.619

---- EQU PICK4

	LOWER	LEVEL	UPPER	MARGINAL
1.1.1	.	.	.	-20.417
1.2.5	.	.	.	-19.899
1.3.9	.	.	.	-31.062
2.1.1	.	.	.	-20.417
2.3.10	.	.	.	-31.062
3.1.1	.	.	.	-20.687
3.2.6	.	.	.	-20.417
4.1.2	.	.	.	-20.104
4.2.6	.	.	.	49.989
5.1.2	.	.	.	-20.104
5.2.7	.	.	.	-20.417
5.3.12	.	.	.	-31.062
6.1.2	.	.	.	EPS
6.2.11	.	.	.	-142.206
7.1.3	.	.	.	-20.147
7.2.7	.	.	.	49.989
8.1.3	.	.	.	-20.104
8.2.8	.	.	.	-20.417
8.3.13	.	.	.	-31.062
9.1.4	.	.	.	-20.620
9.2.8	.	.	.	-22.619

---- EQU PICK5

	LOWER	LEVEL	UPPER	MARGINAL
1.1.1	-INF	.	.	.
1.2.5	-INF	.	.	.
1.3.9	-INF	.	.	.

GAMS 2.25.056 VAX/VMS 05/15/93 17:24:28 PAGE 31
 simple example of aggregate task synthesis
 Solution Report SOLVE AGGTASK USING MIP FROM LINE 1264

EQU PICK5

	LOWER	LEVEL	UPPER	MARGINAL
2.1.1	-INF	.	.	.
2.3.10	-INF	.	.	.
3.1.1	-INF	.	.	51.481
3.2.6	-INF	.	.	0.978
4.1.2	-INF	.	.	.
4.2.6	-INF	.	.	.
5.1.2	-INF	.	.	.
5.2.7	-INF	.	.	0.978
5.3.12	-INF	.	.	.
6.1.2	-INF	.	.	.
6.2.11	-INF	.	.	.
7.1.3	-INF	.	.	.
7.2.7	-INF	.	.	.
8.1.3	-INF	.	.	.
8.2.8	-INF	.	.	11.896
8.3.13	-INF	.	.	.
9.1.4	-INF	.	.	57.723
9.2.8	-INF	.	.	.

---- EQU PICK6

	LOWER	LEVEL	UPPER	MARGINAL
1.1.1	-INF	.	.	.
1.2.5	-INF	.	.	.
1.3.9	-INF	.	.	.
2.1.1	-INF	.	.	.
2.3.10	-INF	.	.	.
3.1.1	-INF	.	.	.
3.2.6	-INF	.	.	.
4.1.2	-INF	.	.	.
4.2.6	-INF	.	.	.
5.1.2	-INF	.	.	.
5.2.7	-INF	.	.	.
5.3.12	-INF	.	.	.
6.1.2	-INF	.	.	.
6.2.11	-INF	.	.	.
7.1.3	-INF	.	.	.
7.2.7	-INF	.	.	.
8.1.3	-INF	.	.	.
8.2.8	-INF	.	.	.
8.3.13	-INF	.	.	.
9.1.4	-INF	.	.	.
9.2.8	-INF	.	.	.

---- EQU PICK7

	LOWER	LEVEL	UPPER	MARGINAL
1.1.1 .1	-INF	.	.	.
1.1.1 .2	-INF	.	.	.
1.2.5 .1	-INF	-1.172E-5	.	.
1.2.5 .2	-INF	.	.	.
1.3.9 .1	-INF	-1.633E-5	.	.
1.3.9 .2	-INF	.	.	.

GAMS 2.25.056 VAX/VMS

05/15/93 17:24:28 PAGE 32

simple example of aggregate task synthesis

Solution Report SOLVE AGGTASK USING MIP FROM LINE 1264

EQU PICK7

	LOWER	LEVEL	UPPER	MARGINAL
2.1.1 .1	-INF	.	.	27.595
2.1.1 .2	-INF	-0.902	.	.
2.3.10.1	-INF	-1.633E-5	.	.
2.3.10.2	-INF	.	.	.
3.1.1 .1	-INF	.	.	EPS
3.1.1 .2	-INF	.	.	.
3.2.6 .1	-INF	.	.	.
3.2.6 .2	-INF	.	.	.
4.1.2 .1	-INF	.	.	.
4.1.2 .2	-INF	.	.	.
4.2.6 .1	-INF	-1.193E-4	.	.
4.2.6 .2	-INF	.	.	.
5.1.2 .1	-INF	.	.	.
5.1.2 .2	-INF	.	.	.
5.2.7 .1	-INF	.	.	.
5.2.7 .2	-INF	.	.	.
5.3.12.1	-INF	-1.633E-5	.	.
5.3.12.2	-INF	.	.	.
6.1.2 .1	-INF	.	.	.
6.1.2 .2	-INF	.	.	.
6.2.11.1	-INF	.	.	.
6.2.11.2	-INF	.	.	.
7.1.3 .1	-INF	.	.	.
7.1.3 .2	-INF	.	.	.
7.2.7 .1	-INF	-1.193E-4	.	.
7.2.7 .2	-INF	.	.	.
8.1.3 .1	-INF	.	.	.
8.1.3 .2	-INF	.	.	.
8.2.8 .1	-INF	.	.	.
8.2.8 .2	-INF	.	.	.
8.3.13.1	-INF	-0.345	.	.
8.3.13.2	-INF	.	.	.
9.1.4 .1	-INF	.	.	.
9.1.4 .2	-INF	.	.	.
9.2.8 .1	-INF	.	.	.
9.2.8 .2	-INF	.	.	.

---- EQU PICK9

	LOWER	LEVEL	UPPER	MARGINAL
1.1.1 .1	-INF	.	.	.
1.2.5 .1	-INF	.	.	.
1.3.9 .1	-INF	.	.	.
2.1.1 .1	-INF	.	.	.
2.3.10.1	-INF	.	.	.
3.1.1 .1	-INF	.	.	.
3.2.6 .1	-INF	.	.	.
4.1.2 .1	-INF	.	.	.
4.2.6 .1	-INF	.	.	.
5.1.2 .1	-INF	.	.	.
5.2.7 .1	-INF	.	.	.
5.3.12.1	-INF	.	.	.
6.1.2 .1	-INF	.	.	.
6.2.11.1	-INF	.	.	.
7.1.3 .1	-INF	.	.	.

GAMS 2.25.056 VAX/VMS 05/15/93 17:24:28 PAGE 33
 simple example of aggregate task synthesis
 Solution Report SOLVE AGGTASK USING MIP FROM LINE 1264

EQU PICK9

	LOWER	LEVEL	UPPER	MARGINAL
7.2.7 .1	-INF	.	.	.
8.1.3 .1	-INF	.	.	.
8.2.8 .1	-INF	.	.	.
8.3.13.1	-INF	.	.	.
9.1.4 .1	-INF	.	.	.
9.2.8 .1	-INF	.	.	.

---- EQU ECOST1

	LOWER	LEVEL	UPPER	MARGINAL
1.1.1 .4	.	.	+INF	-12.920
1.2.5 .1	.	.	+INF	-12.920
1.3.9 .3	.	.	+INF	-12.920
2.1.1 .4	.	.	+INF	-12.920
2.3.10.3	.	.	+INF	-12.920
3.1.1 .1	.	.	+INF	-12.920
3.2.6 .4	.	.	+INF	-12.920
4.1.2 .2	.	.	+INF	-12.920
4.2.6 .4	.	.	+INF	-12.920
5.1.2 .2	.	.	+INF	-12.920
5.2.7 .4	.	.	+INF	-12.920
5.3.12.3	.	.	+INF	-12.920
6.1.2 .4	.	.	+INF	-12.920
6.2.11.3	.	.	+INF	-12.920
7.1.3 .1	.	.	+INF	-12.920
7.2.7 .4	.	.	+INF	-12.920
8.1.3 .1	.	.	+INF	-12.920
8.2.8 .4	.	.	+INF	-12.920
8.3.13.3	.	.	+INF	-12.920
9.1.4 .2	.	.	+INF	-12.920
9.2.8 .4	.	.	+INF	-12.920

---- EQU UCOST1

	LOWER	LEVEL	UPPER	MARGINAL
1.1.1 .4	.	.	+INF	-1.000
1.2.5 .1	.	.	+INF	-1.000
1.3.9 .3	.	.	+INF	-1.000
2.1.1 .4	.	.	+INF	-1.000
2.3.10.3	.	.	+INF	-1.000
3.1.1 .1	.	.	+INF	-1.000
3.2.6 .4	.	.	+INF	-1.000
4.1.2 .2	.	.	+INF	-1.000
4.2.6 .4	.	.	+INF	-1.000
5.1.2 .2	.	.	+INF	-1.000
5.2.7 .4	.	.	+INF	-1.000
5.3.12.3	.	.	+INF	-1.000
6.1.2 .4	.	.	+INF	-1.000
6.2.11.3	.	.	+INF	-1.000
7.1.3 .1	.	.	+INF	-1.000
7.2.7 .4	.	.	+INF	-1.000
8.1.3 .1	.	.	+INF	-1.000
8.2.8 .4	.	.	+INF	-1.000

GAMS 2.25.056 VAX/VMS

05/15/93 17:24:28 PAGE 34

simple example of aggregate task synthesis

Solution Report SOLVE AGGTASK USING MIP FROM LINE 1264

EQU UCOST1

	LOWER	LEVEL	UPPER	MARGINAL
8.3.13.3	.	.	+INF	-1.000
9.1.4 .2	.	.	+INF	-1.000
9.2.8 .4	.	.	+INF	-1.000

---- EQU UCOST2A

	LOWER	LEVEL	UPPER	MARGINAL
1.2.5.1	-INF	.	.	.
4.1.2.2	-INF	.	.	.
5.1.2.2	-INF	.	.	.
7.1.3.1	-INF	.	.	.
8.1.3.1	-INF	.	.	.

---- EQU ECOST2B

	LOWER	LEVEL	UPPER	MARGINAL
1.1.1 .4	-INF	.	.	.
1.2.5 .1	-INF	.	.	.
1.3.9 .3	-INF	.	.	.
2.1.1 .4	-INF	.	.	.
2.3.10.3	-INF	.	.	.
3.1.1 .1	-INF	.	.	.
3.2.6 .4	-INF	.	.	.
4.1.2 .2	-INF	.	.	.
4.2.6 .4	-INF	.	.	.
5.1.2 .2	-INF	.	.	.
5.2.7 .4	-INF	.	.	.
5.3.12.3	-INF	.	.	.
6.1.2 .4	-INF	.	.	.
6.2.11.3	-INF	.	.	.
7.1.3 .1	-INF	.	.	.
7.2.7 .4	-INF	.	.	.
8.1.3 .1	-INF	.	.	.
8.2.8 .4	-INF	.	.	.
8.3.13.2	-INF	.	.	.
9.1.4 .2	-INF	.	.	.
9.2.8 .4	-INF	.	.	.

---- EQU ECOST2C

	LOWER	LEVEL	UPPER	MARGINAL
1.2.5.1	.	.	+INF	-12.920
4.1.2.2	.	.	+INF	-12.920
5.1.2.2	.	.	+INF	-12.920
7.1.3.1	.	.	+INF	-12.920
8.1.3.1	.	.	+INF	-12.920

GAMS 2.25.056 VAX/VMS 05/15/93 17:24:28 PAGE 35
 simple example of aggregate task synthesis
 Solution Report SOLVE AGGTASK USING MIP FROM LINE 1264

---- EQU UCOST2C

	LOWER	LEVEL	UPPER	MARGINAL
1.2.5.1	.	.	+INF	-1.000
4.1.2.2	.	.	+INF	-1.000
5.1.2.2	.	.	+INF	-1.000
7.1.3.1	.	.	+INF	-1.000
8.1.3.1	.	.	+INF	-1.000

---- EQU XUPB

	LOWER	LEVEL	UPPER	MARGINAL
1.2	-INF	-1.000	.	.
4.1	-INF	.	.	.
5.1	-INF	.	.	.
7.1	-INF	.	.	.
8.1	-INF	.	.	.

---- EQU YUPB

	LOWER	LEVEL	UPPER	MARGINAL
1.1	-INF	1.000	1.000	.
1.2	-INF	1.000	1.000	.
1.3	-INF	1.000	1.000	.
2.1	-INF	1.000	1.000	.
2.3	-INF	1.000	1.000	.
3.1	-INF	1.000	1.000	.
3.2	-INF	1.000	1.000	.
4.1	-INF	1.000	1.000	.
4.2	-INF	1.000	1.000	.
5.1	-INF	1.000	1.000	.
5.2	-INF	1.000	1.000	.
5.3	-INF	1.000	1.000	.
6.1	-INF	.	1.000	.
6.2	-INF	.	1.000	.
7.1	-INF	1.000	1.000	.
7.2	-INF	1.000	1.000	.
8.1	-INF	1.000	1.000	.
8.2	-INF	1.000	1.000	.
8.3	-INF	1.000	1.000	.
9.1	-INF	1.000	1.000	.
9.2	-INF	.	1.000	.

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU PCON4		4.000	4.000	+INF -190.588

---- EQU LDEF1

	LOWER	LEVEL	UPPER	MARGINAL
1.2.5.1	-INF	.	.	.
1.2.5.2	-INF	.	.	.
1.2.5.3	-INF	.	.	.
4.1.2.1	-INF	-1.000	.	.

GAMS 2.25.056 VAX/VMS

05/15/93 17:24:28 PAGE 36

simple example of aggregate task synthesis

Solution Report SOLVE AGGTASK USING MIP FROM LINE 1264

EQU LDEF1

	LOWER	LEVEL	UPPER	MARGINAL
4.1.2.2	-INF	-1.000	.	.
4.1.2.3	-INF	.	.	.
5.1.2.1	-INF	-1.000	.	.
5.1.2.2	-INF	-1.000	.	.
5.1.2.3	-INF	.	.	.
7.1.3.1	-INF	-1.000	.	.
7.1.3.2	-INF	-1.000	.	.
7.1.3.3	-INF	.	.	.
8.1.3.1	-INF	-1.000	.	.
8.1.3.2	-INF	-1.000	.	.
8.1.3.3	-INF	.	.	.

---- EQU LDEF2

	LOWER	LEVEL	UPPER	MARGINAL
1.2.5.1	-INF	.	.	.
1.2.5.2	-INF	-1.172E-5	.	.
1.2.5.3	-INF	-1.000	.	.
4.1.2.1	-INF	.	.	.
4.1.2.2	-INF	.	.	.
4.1.2.3	-INF	.	.	.
5.1.2.1	-INF	.	.	.
5.1.2.2	-INF	.	.	.
5.1.2.3	-INF	.	.	.
7.1.3.1	-INF	.	.	.
7.1.3.2	-INF	.	.	.
7.1.3.3	-INF	.	.	.
8.1.3.1	-INF	.	.	.
8.1.3.2	-INF	.	.	.
8.1.3.3	-INF	.	.	.

---- EQU LDEF3

	LOWER	LEVEL	UPPER	MARGINAL
1.2.5.1	-1.000	.	+INF	.
1.2.5.2	-1.000	-1.172E-5	+INF	.
1.2.5.3	-1.000	-1.000	+INF	.
4.1.2.1	-1.000	-1.000	+INF	.
4.1.2.2	-1.000	-1.000	+INF	.
4.1.2.3	-1.000	-1.000	+INF	-59.827
5.1.2.1	-1.000	-1.000	+INF	.
5.1.2.2	-1.000	-1.000	+INF	.
5.1.2.3	-1.000	-1.000	+INF	-59.827
7.1.3.1	-1.000	-1.000	+INF	.
7.1.3.2	-1.000	-1.000	+INF	.
7.1.3.3	-1.000	-1.000	+INF	-87.111
8.1.3.1	-1.000	-1.000	+INF	.
8.1.3.2	-1.000	-1.000	+INF	.
8.1.3.3	-1.000	-1.000	+INF	-80.074

GAMS 2.25.056 VAX/VMS 05/15/93 17:24:28 PAGE 37
 simple example of aggregate task synthesis
 Solution Report SOLVE AGGTASK USING MIP FROM LINE 1264

---- EQU LDEF4

	LOWER	LEVEL	UPPER	MARGINAL
1.2.5.1	-INF	.	1.000	.
1.2.5.2	-INF	1.1715E-5	1.000	.
1.2.5.3	-INF	1.000	1.000	.
4.1.2.1	-INF	1.000	1.000	.
4.1.2.2	-INF	1.000	1.000	.
4.1.2.3	-INF	1.000	1.000	11.467
5.1.2.1	-INF	1.000	1.000	.
5.1.2.2	-INF	1.000	1.000	.
5.1.2.3	-INF	1.000	1.000	11.467
7.1.3.1	-INF	1.000	1.000	.
7.1.3.2	-INF	1.000	1.000	.
7.1.3.3	-INF	1.000	1.000	13.964
8.1.3.1	-INF	1.000	1.000	.
8.1.3.2	-INF	1.000	1.000	.
8.1.3.3	-INF	1.000	1.000	13.415

---- EQU LDEF5

	LOWER	LEVEL	UPPER	MARGINAL
1.2.5.1	-INF	.	.	2.487
1.2.5.2	-INF	.	.	5.964
1.2.5.3	-INF	.	.	11.295
4.1.2.1	-INF	.	.	2.117
4.1.2.2	-INF	.	.	5.087
4.1.2.3	-INF	-1.000	.	.
5.1.2.1	-INF	.	.	2.117
5.1.2.2	-INF	.	.	5.087
5.1.2.3	-INF	-1.000	.	.
7.1.3.1	-INF	.	.	2.487
7.1.3.2	-INF	.	.	5.964
7.1.3.3	-INF	-1.000	.	.
8.1.3.1	-INF	.	.	2.487
8.1.3.2	-INF	.	.	5.964
8.1.3.3	-INF	-1.000	.	.

---- EQU LDEF6

	LOWER	LEVEL	UPPER	MARGINAL
1.2.5.1	.	.	+INF	.
1.2.5.2	.	.	+INF	.
1.2.5.3	.	.	+INF	.
4.1.2.1	.	1.000	+INF	.
4.1.2.2	.	1.000	+INF	.
4.1.2.3	.	.	+INF	.
5.1.2.1	.	1.000	+INF	.
5.1.2.2	.	1.000	+INF	.
5.1.2.3	.	.	+INF	.
7.1.3.1	.	1.000	+INF	.
7.1.3.2	.	1.000	+INF	.
7.1.3.3	.	.	+INF	.
8.1.3.1	.	1.000	+INF	.
8.1.3.2	.	1.000	+INF	.
8.1.3.3	.	.	+INF	.

GAMS 2.25.056 VAX/VMS

05/15/93 17:24:28 PAGE 38

simple example of aggregate task synthesis

Solution Report SOLVE AGGTASK USING MIP FROM LINE 1264

---- EQU LDEF7

	LOWER	LEVEL	UPPER	MARGINAL
1.2.5.1	-INF	.	.	3.356
1.2.5.2	-INF	.	.	83.760
4.1.2.1	-INF	-0.886	.	.
4.1.2.2	-INF	-0.114	.	.
5.1.2.1	-INF	-0.886	.	.
5.1.2.2	-INF	-0.114	.	.
7.1.3.1	-INF	-0.886	.	.
7.1.3.2	-INF	-0.114	.	.
8.1.3.1	-INF	-0.886	.	.
8.1.3.2	-INF	-0.114	.	.

---- EQU LDEF8

	LOWER	LEVEL	UPPER	MARGINAL
1.2.5.1	-INF	-9.654E-6	.	.
1.2.5.2	-INF	-1.000	.	.
4.1.2.1	-INF	.	.	11.305
4.1.2.2	-INF	.	.	401.137
5.1.2.1	-INF	.	.	43.644
5.1.2.2	-INF	.	.	1548.697
7.1.3.1	-INF	.	.	20.200
7.1.3.2	-INF	.	.	778.140
8.1.3.1	-INF	.	.	39.911
8.1.3.2	-INF	.	.	1416.235

---- EQU LDEF9

	LOWER	LEVEL	UPPER	MARGINAL
1.2.5.1	-1.000	-9.654E-6	+INF	.
1.2.5.2	-1.000	-1.000	+INF	.
4.1.2.1	-1.000	-1.000	+INF	.
4.1.2.2	-1.000	-1.000	+INF	.
5.1.2.1	-1.000	-1.000	+INF	.
5.1.2.2	-1.000	-1.000	+INF	.
7.1.3.1	-1.000	-1.000	+INF	.
7.1.3.2	-1.000	-1.000	+INF	.
8.1.3.1	-1.000	-1.000	+INF	.
8.1.3.2	-1.000	-1.000	+INF	.

---- EQU LDEF10

	LOWER	LEVEL	UPPER	MARGINAL
1.2.5.1	-INF	9.6545E-6	1.000	.
1.2.5.2	-INF	1.000	1.000	.
4.1.2.1	-INF	1.000	1.000	0.364
4.1.2.2	-INF	1.000	1.000	12.925
5.1.2.1	-INF	1.000	1.000	0.364
5.1.2.2	-INF	1.000	1.000	12.925
7.1.3.1	-INF	1.000	1.000	1.092
7.1.3.2	-INF	1.000	1.000	42.069
8.1.3.1	-INF	1.000	1.000	0.444

GAMS 2.25.056 VAX/VMS 05/15/93 17:24:28 PAGE 39
 simple example of aggregate task synthesis
 Solution Report SOLVE AGGTASK USING MIP FROM LINE 1264

EQU LDEF10

	LOWER	LEVEL	UPPER	MARGINAL
8.1.3.2	-INF	1.000	1.000	15.754

---- EQU LDEF11

	LOWER	LEVEL	UPPER	MARGINAL
1.2.5.1	-INF	.	.	0.444
1.2.5.2	-INF	.	.	11.080
4.1.2.1	-INF	-0.114	.	.
4.1.2.2	-INF	-0.886	.	.
5.1.2.1	-INF	-0.114	.	.
5.1.2.2	-INF	-0.886	.	.
7.1.3.1	-INF	-0.114	.	.
7.1.3.2	-INF	-0.886	.	.
8.1.3.1	-INF	-0.114	.	.
8.1.3.2	-INF	-0.886	.	.

---- EQU LDEF12

	LOWER	LEVEL	UPPER	MARGINAL
1.2.5.1	.	.	+INF	.
1.2.5.2	.	.	+INF	.
4.1.2.1	.	0.886	+INF	.
4.1.2.2	.	0.114	+INF	.
5.1.2.1	.	0.886	+INF	.
5.1.2.2	.	0.114	+INF	.
7.1.3.1	.	0.886	+INF	.
7.1.3.2	.	0.114	+INF	.
8.1.3.1	.	0.886	+INF	.
8.1.3.2	.	0.114	+INF	.

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU TC1		-INF	.	132.265
---- EQU TC2		-INF	.	59.224
---- EQU TC3		-INF	-1.000	.
---- EQU TC4		-INF	.	132.265
---- EQU TC5		-INF	.	132.265
---- EQU SYM1		-INF	.	.
---- EQU SYM2		-INF	2.000	2.000
---- EQU SYM3		-INF	2.000	2.000
---- EQU SYM5		-INF	.	.
---- EQU SYM6		-INF	.	.
---- EQU SYM7		-INF	-1.000	.
---- EQU OBJ		.	.	1.000

---- VAR PA

	LOWER	LEVEL	UPPER	MARGINAL
1.1.5	.	0.370	+INF	.
1.1.19	.	0.463	+INF	.
1.2.9	.	0.412	+INF	.

GAMS 2.25.056 VAX/VMS

05/15/93 17:24:28 PAGE 40

simple example of aggregate task synthesis

Solution Report SOLVE AGGTASK USING MIP FROM LINE 1264

VAR PA

	LOWER	LEVEL	UPPER	MARGINAL
1.2.10	.	.	+INF	.
1.2.20	.	.	+INF	-99.750
1.2.21	.	0.462	+INF	.
1.3.14	.	0.350	+INF	.
1.3.22	.	0.412	+INF	.
2.1.5	.	0.092	+INF	.
2.1.19	.	0.115	+INF	.
2.3.15	.	0.350	+INF	.
2.3.23	.	0.412	+INF	.
3.1.6	.	0.642	+INF	.
3.1.24	.	0.713	+INF	.
3.2.10	.	0.370	+INF	.
3.2.25	.	0.463	+INF	.
4.1.6	.	0.378	+INF	.
4.1.7	.	0.095	+INF	.
4.1.26	.	0.584	+INF	.
4.1.27	.	.	+INF	.
4.2.10	.	0.041	+INF	.
4.2.25	.	0.052	+INF	.
5.1.6	.	0.378	+INF	.
5.1.7	.	0.095	+INF	.
5.1.26	.	0.584	+INF	.
5.1.39	.	.	+INF	.
5.2.12	.	0.370	+INF	.
5.2.28	.	0.463	+INF	.
5.3.17	.	0.350	+INF	.
5.3.29	.	0.412	+INF	.
6.1.11	.	.	+INF	.
6.1.30	.	.	+INF	.
6.2.16	.	.	+INF	.
6.2.31	.	.	+INF	.
7.1.7	.	0.399	+INF	.
7.1.8	.	0.171	+INF	.
7.1.32	.	0.634	+INF	.
7.1.33	.	.	+INF	.
7.2.12	.	0.041	+INF	.
7.2.34	.	0.052	+INF	.
8.1.7	.	0.368	+INF	.
8.1.8	.	0.158	+INF	.
8.1.32	.	0.584	+INF	.
8.1.40	.	.	+INF	.
8.2.13	.	0.370	+INF	.
8.2.35	.	0.463	+INF	.
8.3.18	.	0.315	+INF	.
8.3.36	.	0.370	+INF	.
9.1.8	.	0.515	+INF	.
9.1.37	.	0.635	+INF	.
9.2.13	.	.	+INF	.
9.2.35	.	.	+INF	.

---- VAR FA

	LOWER	LEVEL	UPPER	MARGINAL
1.1.1	.	0.463	+INF	.
1.2.5	.	0.462	+INF	.

GAMS 2.25.056 VAX/VMS 05/15/93 17:24:28 PAGE 41
 simple example of aggregate task synthesis
 Solution Report SOLVE AGGTASK USING MIP FROM LINE 1264

VAR FA

	LOWER	LEVEL	UPPER	MARGINAL
1.3.9	.	0.412	+INF	.
2.1.1	.	0.115	+INF	.
2.3.10	.	0.412	+INF	.
3.1.1	.	0.713	+INF	.
3.2.6	.	0.463	+INF	.
4.1.2	.	0.584	+INF	.
4.2.6	.	0.052	+INF	.
5.1.2	.	0.584	+INF	.
5.2.7	.	0.463	+INF	.
5.3.12	.	0.412	+INF	.
6.1.2	.	.	+INF	.
6.2.11	.	.	+INF	.
7.1.3	.	0.634	+INF	.
7.2.7	.	0.052	+INF	.
8.1.3	.	0.584	+INF	.
8.2.8	.	0.463	+INF	.
8.3.13	.	0.370	+INF	.
9.1.4	.	0.635	+INF	.
9.2.8	.	.	+INF	.

---- VAR NFA

	LOWER	LEVEL	UPPER	MARGINAL
1	.	1.291	+INF	.
2	.	1.168	+INF	.
3	.	1.218	+INF	.
4	.	0.635	+INF	.
5	.	.	+INF	-4.761E+5
6	.	.	+INF	-1308.775
7	.	.	+INF	-1239.987
8	.	.	+INF	-1448.160
9	.	.	+INF	-6.318E+5
10	.	.	+INF	-6.309E+5
11	.	.	+INF	-5.135E+5
12	.	.	+INF	-6.311E+5
13	.	.	+INF	-6.304E+5
14	.	.	+INF	-7800.195
15	.	.	+INF	-7970.317
16	.	.	+INF	-2374.050
17	.	.	+INF	-8398.377
18	.	.	+INF	-4312.539
19	.	.	+INF	-3632.160
20	.	.	+INF	-3713.500
21	.	.	+INF	-3727.500
22	.	.	+INF	-3524.850
23	.	.	+INF	-3524.850
24	.	.	+INF	-3713.500
25	.	.	+INF	-3632.160
26	.	.	+INF	-3570.000
27	.	.	+INF	-3711.750
28	.	.	+INF	-3632.160
29	.	.	+INF	-3524.850
30	.	.	+INF	-3632.160
31	.	.	+INF	-3524.850
32	.	.	+INF	-3713.500

GAMS 2.25.056 VAX/VMS

05/15/93 17:24:28 PAGE 42

simple example of aggregate task synthesis

Solution Report SOLVE AGGTASK USING MIP FROM LINE 1264

VAR NFA

	LOWER	LEVEL	UPPER	MARGINAL
33	.	.	+INF	-3857.000
34	.	.	+INF	-3632.160
35	.	.	+INF	-3632.160
36	.	.	+INF	-3524.850
37	.	.	+INF	-3570.000
38	.	.	+INF	.
39	.	.	+INF	-3948.000
40	.	.	+INF	-4046.000

---- VAR NPA

	LOWER	LEVEL	UPPER	MARGINAL
1	.	.	+INF	-319.710
2	.	.	+INF	-291.249
3	.	.	+INF	-247.474
4	.	.	+INF	-62.250
5	.	.	+INF	-4773.408
6	.	0.884	+INF	.
7	.	0.442	+INF	.
8	.	0.381	+INF	.
9	.	.	+INF	-6582.920
10	.	.	+INF	-7499.250
11	.	.	+INF	-4511.278
12	.	.	+INF	-7298.619
13	.	.	+INF	-8014.970
14	.	0.350	0.350	2946.789
15	.	0.350	0.350	2742.463
16	.	.	0.450	.
17	.	0.350	0.350	3112.916
18	.	0.315	0.350	.
19	.	0.578	+INF	.
20	.	.	+INF	.
21	.	0.462	+INF	.
22	.	0.412	+INF	.
23	.	0.412	+INF	.
24	.	0.713	+INF	.
25	.	0.515	+INF	.
26	.	1.168	+INF	.
27	.	.	+INF	.
28	.	0.463	+INF	.
29	.	0.412	+INF	.
30	.	.	+INF	.
31	.	.	+INF	.
32	.	1.218	+INF	.
33	.	.	+INF	.
34	.	0.052	+INF	.
35	.	0.463	+INF	.
36	.	0.370	+INF	.
37	.	0.635	+INF	.
38	.	.	+INF	-4147.500
39	.	.	+INF	.
40	.	.	+INF	.

GAMS 2.25.056 VAX/VMS 05/15/93 17:24:28 PAGE 43
 simple example of aggregate task synthesis
 Solution Report SOLVE AGGTASK USING MIP FROM LINE 1264

---- VAR F

	LOWER	LEVEL	UPPER	MARGINAL
1.1.1	.	11.110	+INF	.
1.2.5	.	11.091	+INF	.
1.3.9	.	9.882	+INF	.
2.1.1	.	2.754	+INF	.
2.3.10	.	9.882	+INF	.
3.1.1	.	17.120	+INF	.
3.2.6	.	11.110	+INF	.
4.1.2	.	15.770	+INF	.
4.2.6	.	1.243	+INF	.
5.1.2	.	15.770	+INF	.
5.2.7	.	11.110	+INF	.
5.3.12	.	9.882	+INF	.
6.1.2	.	.	+INF	.
6.2.11	.	.	+INF	.
7.1.3	.	17.120	+INF	.
7.2.7	.	1.243	+INF	.
8.1.3	.	15.770	+INF	.
8.2.8	.	11.110	+INF	.
8.3.13	.	8.888	+INF	.
9.1.4	.	15.251	+INF	.
9.2.8	.	.	+INF	.

---- VAR EQUIP1

	LOWER	LEVEL	UPPER	MARGINAL
1.1	.	15.314	+INF	.
1.2	.	14.036	+INF	.
1.3	.	17.868	+INF	.
2.1	.	7.700	+INF	.
2.3	.	17.868	+INF	.
3.1	.	18.782	+INF	.
3.2	.	15.314	+INF	.
4.1	.	16.567	+INF	.
4.2	.	5.871	+INF	.
5.1	.	16.567	+INF	.
5.2	.	15.314	+INF	.
5.3	.	17.868	+INF	.
6.1	.	.	+INF	.
6.2	.	.	+INF	.
7.1	.	18.782	+INF	.
7.2	.	5.871	+INF	.
8.1	.	17.744	+INF	.
8.2	.	15.314	+INF	.
8.3	.	16.787	+INF	.
9.1	.	16.183	+INF	.
9.2	.	.	+INF	.

---- VAR UTIL1

	LOWER	LEVEL	UPPER	MARGINAL
1.1	.	1035.467	+INF	.
1.2	.	291.198	+INF	.
1.3	.	103.654	+INF	.

GAMS 2.25.056 VAX/VMS

05/15/93 17:24:28 PAGE 44

simple example of aggregate task synthesis

Solution Report SOLVE AGGTASK USING MIP FROM LINE 1264

VAR UTIL1

	LOWER	LEVEL	UPPER	MARGINAL
2.1	.	272.044	+INF	.
2.3	.	103.654	+INF	.
3.1	.	429.365	+INF	.
3.2	.	1035.467	+INF	.
4.1	.	351.650	+INF	.
4.2	.	137.727	+INF	.
5.1	.	351.650	+INF	.
5.2	.	1035.467	+INF	.
5.3	.	103.654	+INF	.
6.1	.	.	+INF	.
6.2	.	.	+INF	.
7.1	.	383.897	+INF	.
7.2	.	137.727	+INF	.
8.1	.	356.515	+INF	.
8.2	.	1035.467	+INF	.
8.3	.	96.350	+INF	.
9.1	.	381.409	+INF	.
9.2	.	.	+INF	.

---- VAR EQUIP2

	LOWER	LEVEL	UPPER	MARGINAL
1.1	.	.	+INF	EPS
1.2	.	.	+INF	.
1.3	.	.	+INF	EPS
2.1	.	.	+INF	EPS
2.3	.	.	+INF	EPS
3.1	.	.	+INF	EPS
3.2	.	.	+INF	EPS
4.1	.	0.888	+INF	.
4.2	.	.	+INF	EPS
5.1	.	0.888	+INF	.
5.2	.	.	+INF	EPS
5.3	.	.	+INF	EPS
6.1	.	.	+INF	EPS
6.2	.	.	+INF	EPS
7.1	.	1.081	+INF	.
7.2	.	.	+INF	EPS
8.1	.	1.038	+INF	.
8.2	.	.	+INF	EPS
8.3	.	.	+INF	EPS
9.1	.	.	+INF	EPS
9.2	.	.	+INF	EPS

---- VAR UTIL2

	LOWER	LEVEL	UPPER	MARGINAL
1.2	.	.	+INF	.
4.1	.	11.489	+INF	.
5.1	.	11.489	+INF	.
7.1	.	15.202	+INF	.
8.1	.	14.003	+INF	.

GAMS 2.25.056 VAX/VMS 05/15/93 17:24:28 PAGE 45
 simple example of aggregate task synthesis
 Solution Report SOLVE AGGTASK USING MIP FROM LINE 1264

---- VAR LAM

	LOWER	LEVEL	UPPER	MARGINAL
1.1.1 .1	.	.	1.000	-11.546
1.1.1 .2	.	.	1.000	.
1.1.1 .3	.	1.000	1.000	39.141
1.2.5 .1	.	.	1.000	-14.073
1.2.5 .2	.	1.1715E-5	1.000	.
1.2.5 .3	.	1.000	1.000	.
1.3.9 .1	.	.	1.000	-62.011
1.3.9 .2	.	1.6326E-5	1.000	.
1.3.9 .3	.	1.000	1.000	.
2.1.1 .1	.	0.098	1.000	.
2.1.1 .2	.	0.902	1.000	.
2.1.1 .3	.	.	1.000	.
2.3.10.1	.	.	1.000	-62.011
2.3.10.2	.	1.6326E-5	1.000	.
2.3.10.3	.	1.000	1.000	.
3.1.1 .1	.	.	1.000	-52.661
3.1.1 .2	.	.	1.000	.
3.1.1 .3	.	1.000	1.000	372.480
3.2.6 .1	.	.	1.000	-20.059
3.2.6 .2	.	.	1.000	.
3.2.6 .3	.	1.000	1.000	115.759
4.1.2 .1	.	.	1.000	-4.156
4.1.2 .2	.	.	1.000	.
4.1.2 .3	.	1.000	1.000	.
4.2.6 .1	.	.	1.000	-21.606
4.2.6 .2	.	1.1932E-4	1.000	.
4.2.6 .3	.	1.000	1.000	.
5.1.2 .1	.	.	1.000	-8.806
5.1.2 .2	.	.	1.000	-4.650
5.1.2 .3	.	1.000	1.000	.
5.2.7 .1	.	.	1.000	-20.059
5.2.7 .2	.	.	1.000	.
5.2.7 .3	.	1.000	1.000	115.759
5.3.12.1	.	.	1.000	-62.011
5.3.12.2	.	1.6326E-5	1.000	.
5.3.12.3	.	1.000	1.000	.
6.1.2 .1	.	.	1.000	-24.792
6.1.2 .2	.	.	1.000	-33.288
6.1.2 .3	.	.	1.000	.
6.2.11.1	.	.	1.000	-22.649
6.2.11.2	.	.	1.000	.
6.2.11.3	.	.	1.000	.
7.1.3 .1	.	.	1.000	-10.610
7.1.3 .2	.	.	1.000	-5.100
7.1.3 .3	.	1.000	1.000	.
7.2.7 .1	.	.	1.000	-21.606
7.2.7 .2	.	1.1932E-4	1.000	.
7.2.7 .3	.	1.000	1.000	.
8.1.3 .1	.	.	1.000	-10.485
8.1.3 .2	.	.	1.000	-4.349
8.1.3 .3	.	1.000	1.000	.
8.2.8 .1	.	.	1.000	-29.157
8.2.8 .2	.	.	1.000	.
8.2.8 .3	.	1.000	1.000	145.274
8.3.13.1	.	.	1.000	-62.011
8.3.13.2	.	0.345	1.000	.
8.3.13.3	.	0.655	1.000	.

GAMS 2.25.056 VAX/VMS

05/15/93 17:24:28 PAGE 46

simple example of aggregate task synthesis

Solution Report SOLVE AGGTASK USING MIP FROM LINE 1264

VAR LAM

	LOWER	LEVEL	UPPER	MARGINAL
9.1.4 .1	.	.	1.000	-56.133
9.1.4 .2	.	.	1.000	.
9.1.4 .3	.	1.000	1.000	327.111
9.2.8 .1	.	.	1.000	-8.605
9.2.8 .2	.	.	1.000	.
9.2.8 .3	.	.	1.000	.

---- VAR LAM2

	LOWER	LEVEL	UPPER	MARGINAL
1.1.1 .1	.	7.5401E-6	1.000	.
1.1.1 .2	.	1.000	1.000	.
1.2.5 .1	.	9.6545E-6	1.000	.
1.2.5 .2	.	1.000	1.000	.
1.3.9 .1	.	2.5713E-5	1.000	.
1.3.9 .2	.	1.000	1.000	.
2.1.1 .1	.	0.788	1.000	.
2.1.1 .2	.	0.212	1.000	.
2.3.10.1	.	2.5713E-5	1.000	.
2.3.10.2	.	1.000	1.000	.
3.1.1 .1	.	9.6270E-6	1.000	.
3.1.1 .2	.	1.000	1.000	.
3.2.6 .1	.	7.5401E-6	1.000	.
3.2.6 .2	.	1.000	1.000	.
4.1.2 .1	.	0.114	1.000	.
4.1.2 .2	.	0.886	1.000	.
4.2.6 .1	.	8.9956E-5	1.000	.
4.2.6 .2	.	1.000	1.000	.
5.1.2 .1	.	0.114	1.000	.
5.1.2 .2	.	0.886	1.000	.
5.2.7 .1	.	7.5401E-6	1.000	.
5.2.7 .2	.	1.000	1.000	.
5.3.12.1	.	2.5713E-5	1.000	.
5.3.12.2	.	1.000	1.000	.
6.1.2 .1	.	.	1.000	-19.785
6.1.2 .2	.	.	1.000	.
6.2.11.1	.	.	1.000	.
6.2.11.2	.	.	1.000	-109.129
7.1.3 .1	.	0.114	1.000	.
7.1.3 .2	.	0.886	1.000	.
7.2.7 .1	.	8.9956E-5	1.000	.
7.2.7 .2	.	1.000	1.000	.
8.1.3 .1	.	0.114	1.000	.
8.1.3 .2	.	0.886	1.000	.
8.2.8 .1	.	7.5401E-6	1.000	.
8.2.8 .2	.	1.000	1.000	.
8.3.13.1	.	0.201	1.000	.
8.3.13.2	.	0.799	1.000	.
9.1.4 .1	.	6.7794E-6	1.000	.
9.1.4 .2	.	1.000	1.000	.
9.2.8 .1	.	.	1.000	.
9.2.8 .2	.	.	1.000	-7.022

GAMS 2.25.056 VAX/VMS 05/15/93 17:24:28 PAGE 47
 simple example of aggregate task synthesis
 Solution Report SOLVE AGTASK USING MIP FROM LINE 1264

---- VAR LAM3

	LOWER	LEVEL	UPPFR	MARGINAL
1.2.5.1	.	.	1.000	-0.642
1.2.5.2	.	.	1.000	-3.853
1.2.5.3	.	.	1.000	-14.245
3.1.1.1	.	.	+INF	EPS
3.1.1.2	.	.	+INF	EPS
3.1.1.3	.	.	+INF	EPS
4.1.2.1	.	.	1.000	-1.897
4.1.2.2	.	.	1.000	-11.381
4.1.2.3	.	1.000	1.000	.
5.1.2.1	.	.	1.000	-1.897
5.1.2.2	.	.	1.000	-11.381
5.1.2.3	.	1.000	1.000	.
7.1.3.1	.	.	1.000	-2.544
7.1.3.2	.	.	1.000	-15.265
7.1.3.3	.	1.000	1.000	.
8.1.3.1	.	.	1.000	-2.539
8.1.3.2	.	.	1.000	-15.233
8.1.3.3	.	1.000	1.000	.
9.1.4.1	.	.	+INF	EPS
9.1.4.2	.	.	+INF	EPS
9.1.4.3	.	.	+INF	EPS

---- VAR LAM4

	LOWER	LEVEL	UPPER	MARGINAL
1.2.5.1	.	.	1.000	.
1.2.5.2	.	1.1715E-5	1.000	.
1.2.5.3	.	1.000	1.000	.
4.1.2.1	.	.	1.000	.
4.1.2.2	.	.	1.000	.
4.1.2.3	.	.	1.000	.
5.1.2.1	.	.	1.000	.
5.1.2.2	.	.	1.000	.
5.1.2.3	.	.	1.000	.
7.1.3.1	.	.	1.000	.
7.1.3.2	.	.	1.000	.
7.1.3.3	.	.	1.000	.
8.1.3.1	.	.	1.000	.
8.1.3.2	.	.	1.000	.
8.1.3.3	.	.	1.000	.

---- VAR LAM5

	LOWER	LEVEL	UPPER	MARGINAL
1.2.5.1	.	.	1.000	.
1.2.5.2	.	.	1.000	.
4.1.2.1	.	0.114	1.000	.
4.1.2.2	.	0.886	1.000	.
5.1.2.1	.	0.114	1.000	.
5.1.2.2	.	0.886	1.000	.
7.1.3.1	.	0.114	1.000	.
7.1.3.2	.	0.886	1.000	.
8.1.3.1	.	0.114	1.000	.

GAMS 2.25.056 VAX/VMS

05/15/93 17:24:28 PAGE 48

simple example of aggregate task synthesis

Solution Report SOLVE AGGTASK USING MIP FROM LINE 1264

VAR LAM5

	LOWER	LEVEL	UPPER	MARGINAL
8.1.3.2	.	0.886	1.000	.

---- VAR LAM6

	LOWER	LEVEL	UPPER	MARGINAL
1.2.5.1	.	9.6545E-6	1.000	.
1.2.5.2	.	1.000	1.000	.
4.1.2.1	.	.	1.000	.
4.1.2.2	.	.	1.000	.
5.1.2.J	.	.	1.000	.
5.1.2.2	.	.	1.000	.
7.1.3.1	.	.	1.000	.
7.1.3.2	.	.	1.000	.
8.1.3.1	.	.	1.000	.
8.1.3.2	.	.	1.000	.

---- VAR Y

	LOWER	LEVEL	UPPER	MARGINAL
1.1	.	1.000	1.000	EPS
1.2	.	1.000	1.000	EPS
1.3	.	1.000	1.000	EPS
2.1	.	1.000	1.000	-70.771
2.3	.	1.000	1.000	132.265
3.1	.	1.000	1.000	3.533
3.2	.	1.000	1.000	-46.700
4.1	.	1.000	1.000	232.996
4.2	.	1.000	1.000	EPS
5.1	.	1.000	1.000	232.996
5.2	.	1.000	1.000	85.565
5.3	.	1.000	1.000	EPS
6.1	.	.	1.000	-28.586
6.2	.	.	1.000	EPS
7.1	.	1.000	1.000	420.105
7.2	.	1.000	1.000	EPS
8.1	.	1.000	1.000	377.822
8.2	.	1.000	1.000	96.483
8.3	.	1.000	1.000	EPS
9.1	.	1.000	1.000	9.842
9.2	.	.	1.000	-56.334

---- VAR X

	LOWER	LEVEL	UPPER	MARGINAL
1.2	.	.	1.000	87.116
4.1	.	1.000	1.000	-84.583
5.1	.	1.000	1.000	-84.583
7.1	.	1.000	1.000	-144.236
8.1	.	1.000	1.000	-109.686

GAMS 2.25.056 VAX/VMS 05/15/93 17:24:28 PAGE 49
 simple example of aggregate task synthesis
 Solution Report SOLVE AGGTASK USING MIP FROM LINE 1264

---- VAR DEL

	LOWER	LEVEL	UPPER	MARGINAL
1.1.1 .1	.	.	1.000	EPS
1.1.1 .2	.	1.000	1.000	EPS
1.2.5 .1	.	.	1.000	25.880
1.2.5 .2	.	1.000	1.000	EPS
1.3.9 .1	.	.	1.000	EPS
1.3.9 .2	.	1.000	1.000	-64.797
2.1.1 .1	.	1.000	1.000	EPS
2.1.1 .2	.	.	1.000	50.688
2.3.10.1	.	.	1.000	EPS
2.3.10.2	.	1.000	1.000	-64.797
3.1.1 .1	.	.	1.000	EPS
3.1.1 .2	.	1.000	1.000	EPS
3.2.6 .1	.	.	1.000	EPS
3.2.6 .2	.	1.000	1.000	EPS
4.1.2 .1	.	.	1.000	-4.650
4.1.2 .2	.	1.000	1.000	EPS
4.2.6 .1	.	.	1.000	22.727
4.2.6 .2	.	1.000	1.000	EPS
5.1.2 .1	.	.	1.000	EPS
5.1.2 .2	.	1.000	1.000	EPS
5.2.7 .1	.	.	1.000	EPS
5.2.7 .2	.	1.000	1.000	EPS
5.3.12.1	.	.	1.000	EPS
5.3.12.2	.	1.000	1.000	-64.797
6.1.2 .1	.	.	1.000	EPS
6.1.2 .2	.	.	1.000	EPS
6.2.11.1	.	.	1.000	EPS
6.2.11.2	.	.	1.000	EPS
7.1.3 .1	.	.	1.000	EPS
7.1.3 .2	.	1.000	1.000	EPS
7.2.7 .1	.	.	1.000	22.727
7.2.7 .2	.	1.000	1.000	EPS
8.1.3 .1	.	.	1.000	EPS
8.1.3 .2	.	1.000	1.000	EPS
8.2.8 .1	.	.	1.000	EPS
8.2.8 .2	.	1.000	1.000	EPS
8.3.13.1	.	.	1.000	EPS
8.3.13.2	.	1.000	1.000	-64.797
9.1.4 .1	.	.	1.000	EPS
9.1.4 .2	.	1.000	1.000	EPS
9.2.8 .1	.	.	1.000	EPS
9.2.8 .2	.	.	1.000	EPS

---- VAR DEL2

	LOWER	LEVEL	UPPER	MARGINAL
1.1.1 .1	.	1.000	1.000	EPS
1.2.5 .1	.	1.000	1.000	EPS
1.3.9 .1	.	1.000	1.000	EPS
2.1.1 .1	.	1.000	1.000	EPS
2.3.10.1	.	1.000	1.000	EPS
3.1.1 .1	.	1.000	1.000	EPS
3.2.5 .1	.	1.000	1.000	EPS
4.1.2 .1	.	1.000	1.000	EPS
4.2.6 .1	.	1.000	1.000	-67.880

GAMS 2.25.056 VAX/VMS 05/15/93 17:24:28 PAGE 50
 simple example of aggregate task synthesis
 Solution Report SOLVE AGGTASK USING MIP FROM LINE 1264

VAR DEL2

	LOWER	LEVEL	UPPER	MARGINAL
5.1.2 .1	.	1.000	1.000	EPS
5.2.7 .1	.	1.000	1.000	EPS
5.3.12.1	.	1.000	1.000	EPS
6.1.2 .1	.	.	1.000	EPS
6.2.11.1	.	.	1.000	EPS
7.1.3 .1	.	1.000	1.000	EPS
7.2.7 .1	.	1.000	1.000	-67.880
8.1.3 .1	.	1.000	1.000	EPS
8.2.8 .1	.	1.000	1.000	EPS
8.3.13.1	.	1.000	1.000	EPS
9.1.4 .1	.	1.000	1.000	EPS
9.2.8 .1	.	.	1.000	EPS

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR Z	-INF	4513.119	+INF	.

**** REPORT SUMMARY :
 0 NONOPT
 0 INFEASIBLE
 0 UNBOUNDED

EXECUTION TIME = 1.670 SECONDS VERID VMS-00-056

USER: Erwin GAMS G921024-2037CX-VMS

**** FILE SUMMARY

INPUT SYS\$USER2:[D232JK60]BZNTRL_DVL6.GMS
 OUTPUT SYS\$USER2:[D232JK60]BZNTRL_DVL6.LIS

Appendix 5: Code for Batch Distillation Module

A5.1. Computational Code for Batch Distillation Column Module

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C QUICK_CALC (FORTRAN 77)
C This program simulates the separation of a binary
C mixture via batch distillation. The method developed
C Diwekar and Madhavan (1991) is used. In essence, the still
C is modeled as a continuous rectifier with time-varying
C feed. The method is a little more complex than the following
C description, but roughly, we proceed as follows:
C
C   Min no. trays found by Hengstebeck-Geddes eq.
C   Min reflux ratio found from Underwood eqs.
C   No. Trays found from Gilliland Correlation
C
C           then
C
C   (1) Still composition of light component is reduced by an
C       incremental value.
C   (2) System equilibrium is recomputed (the equation
C       for finding this new equilibrium point is the main contribution
C       of the paper
C   (3) Iterations proceed until still or overhead targets
C       are reached. At that point a slop cut is made if
C       the still composition has not yet been reached.
C
C The program is totally self contained; there are no input
C files.
C
C coded by jonathan knight
C dept. chemical engineering
C MIT
C
C created Feb 16, 1993
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

C *** THIS IS A TRIVIAL LITTLE DRIVER PROGRAM
C       program quick_calc

C
C *** VARIABLES
C
C ** NO. COMPONENTS
C       integer*4 nc

C ** NO. TRAYS
C       integer*4 nn

C ** REFERENCE COMPONENT (I.E., THE DISTILLATE)
C       integer*4 reference

C ** RELATIVE VOLATILITIES
C       real*8 a(2)
```



```
C ** REFLUX RATIO
  real*8 rr

C ** INSTANTANEOUS STILL COMPOSITION
  real*8 xb(2)

C ** INSTANTANEOUS DISTILLATE COMPOSITION
  real*8 xd(2)

C ** MOVING AVERAGE DISTILLATE COMPOSITION
  real*8 xavg

C ** TOTAL LB-MOLES OF DISTILLATE
  real*8 dst_total

C ** TARGET STILL COMPOSITION
  real*8 xbr_target

C ** TARGET AVERAGE DISTILLATE COMPOSITION
  real*8 xavg_target

C ** LB-MOLES OF COMPONENTS IN THE SLOP CUTS
  real*8 slop(2)

C ** LB-MOLES IN STILL
  real*8 btm_new

C ** SPECIFICATION (I.E., A GUESS AT A REALISTIC VALUE)
  FOR INITIAL DISTILLATE COMPOSITION
  real*8 xdstart
  xdstart = 0.2

C ***HERE WE CREATE A LOOP WHERE WE VARY THE INITIAL
  DISTILLATE COMPOSITION

  do jjj=1,14

C ***THE COMPUTATIONAL CODE BELOW DOES NOT REZERO OR REINITIALIZE
  C ANYTHING, SO WE NEED TO RESET THESE VARIABLES ON
  C EACH ITERATION. TAKE NOTE OF THESE PARAMS BECAUSE
  C THEY ARE IMPORTANT

C   GENERAL SPECS
  nc = 2
  nn=0
  rr = 3.0
  reference=1

C   THERMODYNAMIC AND MOLE FRACTION SPECS
  a(1)=2.5
  a(2)=1.
  xb(1)=.2
  xb(2)=.8

C   DESIGN TARGETS
  xavg_target=0.2
  xbr_target=0.05

C   NET MATERIAL SPECS
  slop(1) = 0.
```

```

        slop(2) = 0.
        xavg =0.
        btm_new=1.0
        dst_total=0.

C      UPDATES
        xdstart = xdstart + 0.05
        xd(1) = xdstart

C *** CALL THE MAIN PROGRAM
        call get_fraction(a,reference,rr,xb,xd,nc,nn,
&          xbr_target,xavg_target,
&          btm_new,xavg,dst_total,slop)

C *** PRINT OUT THE OVERALL RESULTS
        write(*,*) ' '
        write(*,'(A24,10x,f12.7)') '---INITIAL DISTILLATE X:', xdstart
        write(*,'(7x,A17,10x,f12.7)') 'total distillate:', dst_total
        write(*,'(10x,A14,10x,f12.7)') 'total bottoms:', btm_new
        write(*,'(18x,A6,10x,f12.7,10x,f12.7)') 'slops:', slop(1), slop(2)
        write(*,'(19x,A5,10x,f12.7)') 'xavg:', xavg
        write(*,'(20x,A4,10x,f12.7,10x,f12.7)') 'xbs:', xb(1), xb(2)
        write(*,'(20x,A4,10x,f12.7,10x,f12.7)') 'xds:', xd(1), xd(2)
        write(*,'(21x,A3,10x,I4)') 'nn:',nn
        write(*,'(21x,A3,10x,f12.7)') 'rr:',rr
        write(*,'(A50)') '-----'
        write(*,'(20x,A12,10x,A12)') ' xb(light) ', xavg

        end do
        stop
        end

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C GET_FRACTION (FORTRAN 77)
C Computes the time, yield and mass flow of a fraction for a
C mixture being separated by a batch distillation column. This
C module has a design mode which is activated for the initial
C fraction.
C
C coded by and property of jonathan knight
C dept. chemical engineering
C MIT
C
C created Feb 16
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        subroutine get_fraction(a,referenceg,rr,xb,xd,nc,nn,
&          xbr_target,xavg_target,
&          btm_new,xavg,dst_total,slop)
C
C *** ARGS
C
        integer*4 nc,nn,referenceg
        real*8 a(*),rr,xb(*),xd(*),xavg,dst_total,xbr_target,
& xavg_target,slop(*),btm_new

C
C *** LOCALS
C
        real*8 dxbr, xbold(2)
        real*8 dbtm,btm_old,dst_new,prd_total
        real*8 phi,rmin,c1

```

```
integer*4 i,j,k,reference

C ** OVERHEAD/BTMS COMPOSITION,VOLATILITY OF REFERENCE COMPONENT
real*8 xbr,xdr,ar

C ** MAXIMUM NUMBER OF COMPOSITION SETTING ITERATIONS
integer*4 maxit

C ** INTIAL POUND MOLES TO STILL (USE IN DISTILLATE MAT. BALANCE.
real*8 dst_start

C ** SLOP STREAM
real*8 slop_new

C ***INITIALIZE REFERENCE COMPONENT AND INITIAL POUND MOLES
C FED TO STILL
reference = referenceg
dst_start=btm_new

C
C ***THESE ARE SOME INITIALIZATION CALCULATIONS TO GET
C COLUMN SPECS BEFORE WE GO INTO THE MAIN LOOP
C
C ***BASED ON INITIAL DISTILLATE COMPOSITION CALCULATE A
C MINIMUM NUMBER OF TRAYS
ar = a(reference)
call get_c1(nc,a,xb,xd(reference),
& xb(reference),a(reference),c1)
C ***GET THE DISTILLATE COMPOSITION OF THE REMAINING
C COMPONENTS
call get_xd(a,c1,nc,xb,xd,xd(reference),
& xb(reference),a(reference),reference)
C *** DO THE UNDERWOOD EQS.
call get_phi(a,xb,nc,reference,phi)
call get_rmin(nc,a,xd,phi,rmin)

C ***IF YOUR CHOICE OF R IS TOO LOW, THEN RESET R USING RMIN
if(rmin .gt. rr) rr = 1.32*rmin

C ***IF THE TRAYS HAVE NOT ALREADY BEEN SPECIFIED, THEN
C COMPUTE THE NUMBER FROM GILLILANDS CORRELATION
if(nn .eq. 0) call get_trays(rr,rmin,c1,nn)

C
C ***GET READY TO ENTER MAIN LOOP (WE START WITH THE INITIAL
C LIGHT COMPONENT MOLE FRACTION AND DECREMENT IT ITERATIVELY,
C RECOMPUTING THE COLUMN EQUILIBRIUM AT EACH STEP):
C *SET XAVG TO ITS UPPER BOUND (COMPUTED CORRECTLY ON
C FIRST ITERATION)
C *SET AN UPPER BOUND ON THE NUMBER OF INCREMENTS OF
C THE LIGHT COMPONENT STILL COMPOSITION
C *SET THE DECREMENT ON THE LIGHT COMPONENT STILL COMPOSITION
C *INITIALIZE AMOUNT OF PRODUCT PRODUCED 'SO FAR'
```

```
C    *INITIALIZE THE ITERATION COUNTER TO ZERO
C
C        xavg=1.0
C        maxit = 10000
C        dxbr = 5.0E-3
C        prd_total=0.0
C        i=0
C
C    ***ZERO OUT SLOP CUT
C        do 5444 j=1,nc
C            slop(i)=0.0
5444 continue
C
C
C    ***START ITERATIONS USING A LOW-CLASS IF LOOP
C    KEEP ITERATING UNTIL STILL COMPOSITION HAS
C    BEEN ACHIEVED
1100 if(xb(reference) .gt. xbr_target .and.
C        & i .le. maxit) then
C            i=i+1
C
C    ***DECREMENT COMPOSITION OF LIGHT COMPONENT IN STILL
C        xb(reference) = xb(reference) -dxbr
C
C    ***REMEMBER THE OLD STILL COMPOSITIONS FROM PREVIOUS
C    ITERATION. THEY ARE NEEDED BY THE ALGORITHM
C        do j=1,nc
C            if(j .ne. reference) then
C                xbold(j) = xb(j)
C            endif
C        end do
C
C    ***RECOMPUTE THE COLUMN EQUILIBRIUM. SEE THE PAPER
C    BY DIWEKAR AND MADHAVAN (1991). I DIDN'T FEEL LIKE
C    BEING SOPHISTICATED IN THE CONVERGENCE SCHEME, SO
C    I JUST SET AN UPPER BOUND OF 50 MAJOR INTERATIONS.
C    THIS IS MORE THAN ENOUGH
C
C        do 311 k=1,50
C            xdr = xd(reference)
C            call get_xb(C1,nc,xb,xbold,dxbr,xd(reference),
C            &        xb(reference),ar,reference,a)
C            call get_phi(a,xb,nc,reference,phi)
C            call hg_calc(a,c1,nc,mn,phi,rr,xb,xd,k,
C            &        xd(reference),xb(reference),ar,reference)
C            call get_xd(a,C1,nc,xb,xd,xd(reference),
C            &        xb(reference),ar,reference)
311 continue
C
C    ***NOW THAT WE KNOW THE NEW EQUILIBRIUM AND LET'S COMPUTE
C    THE MATERIAL FLOWS THAT OCCURRED
C        xbr = xb(reference)
C        xdr = xd(reference)
C    *THIS IS A DIFFERENTIAL MATERIAL BALANCE WRITTEN UNDER THE
C    ASSUMPTION THAT THE DISTILLATE COMPOSITION IS CONSTANT
C    IN THE DIFFERENTIAL INCREMENT
```

```

        dbtm = exp(-dxbr/(xdr-xbr))
        btm_old=btm_new
        btm_new = btm_new*dbtm

C ***WHILE THE AVERAGE VALUE IS IN THE ACCEPTIBLE RANGE WE
C   COMPUTE IT.  AFTERWARDS, WE MAKE SLOP CUT.
        if (xavg .ge. xavg_target) then
            dst_new = btm_old-btm_new
            dst_total = dst_start-btm_new
            prd_total = prd_total+xdr*dst_new
            xavg = prd_total/dst_total
        else
            slop_new = btm_old-btm_new
            do 232 k=1,nc
                slop(k)=slop(k)+slop_new*xd(k)
232         continue
            endif

C ***PRINT OUT THE INTERMEDIATE MOVING AVERAGES
        write(*,'(20x,f12.7,10x,f12.7)') xb(1),xavg

        goto 1100
    endif
C
    return
end

C subroutine to compute still pot compositions, xb(i), i>1
C
C coded by : jonathan knight
C dept. che, MIT
C Jan 5, 1992
C
C Inputs are
C   C1 - dp hg constant
C   nc - i*4 number of components
C   xb - dp array of still pot boiling points
C   dxbl - change in bottom still pot composition of comp. 1
C   xdl - distillate composition of comp. 1
C   a - separation factors
C
C Outputs are
C   xb(i), i>1.

        subroutine get_xb(C1,nc,xb,xbold,dxbr,xdr,
            & xbr,ar,reference,a)
#include "equip_params.f"

C
C *** args
C
        integer*4 nc,reference
        real*8 C1, xb(*), a(*), dxbr, xdr, xbr,xbold(*),
            & ar

C
C *** locals
C
        integer*4 i
        real*8 sum

```

```

C
C
C
C
    sum = 0.0
    do i=1,nc
      if(i .ne. reference) then
        xb(i) = xbold(i) *
&    exp(dxbr/(xdr-xbr)*((a(i)/ar)**C1*xdr/xbr - 1.0))
        sum = sum + xb(i)
      endif
    end do
C
    do i=1,nc
      if(i .ne. reference) then
        xb(i) = xb(i)/sum * (1.0-xbr)
      endif
    end do
C
    return
    end

C
C Simple program using newtons method for computing Hengstebeck-Geddes
C constant and light component overhead fraction.
C
C CODED BY: Jonathan Knight
C Dept Chem. Engng., MIT
C January 4, 1992
C
C Inputs are:
C   nc - I*4 number of components
C   NN - I*4 number of trays
C   RR - dp reflux ratio
C   phi - dp Underwood Equation constant
C   a - dp array of separation factors
C   xb - dp array of bottom compositions
C   xd - dp array of overhead compositions (first value is an old value)
C   C1 - dp initial guess of H-G constant
C
C Outputs:
C   xd(1), C1
C
C Notes:
C   Jan 4: 25 set as upper limit on number components in mixture.
C
    subroutine hg_calc(a,c1,nc,nn,phi,rr,xb,xd,k,
&    xdr,xbr,ar,reference)
#include "equip_params.f"
C
C *** args
C
    integer*4 nc,nn,k,reference
    real*8 a(*),c1,phi,rr,xb(*),xd(*),xdr,xbr,ar

C
C *** locals
C
    (equations whose roots are to be found. see Diwekar., IECR, 1991, 713.)

```

```

      real*8 G1, G2

      integer*4 i

C      (derivatives for newtons method)
      real*8 G1x, G1C, G2x, G2C

C      differential change
      real*8 dx,dC

C      old values and error norm
      real*8 xold, Cold, err1, err2
C
C
C      initialize error norms to large values
      err1 = 1.0E6
      err2 = 1.0E6
      i=1

1000 if(i .lt. 100 .and. (err1 .gt. 1.0E-6 .or.
&          err2 .gt. 1.0E-6)) then

C      make an evaluation of functions and derivatives
      call hg_eval(a,C1,nc,nn,phi,rr,xb,xd,G1,G2,G1x,G1C,
&          G2x,G2C,k,xdr,xbr,ar,reference)

C      save old values of xdl and C1 (only to compute error norm)
      xold = xdr
      Cold = C1

C      compute differential changes

      adj = G1x*G2C - G2x*G1C

      dx = -(G1*G2C - G2*G1C) / adj
      dC = -(G2*G1x - G1*G2x) / adj

C      update values
      xdr = xdr + dx
      C1 = C1 + dC

C      compute error norms
      err1 = sqrt(G1**2 + G2**2)
      err2 = sqrt((xdr-xold)**2+(C1-Cold)**2)

C      print out values

C      program control
      i=i+1
      goto 1000
      endif
      return
      end

C
C Returns values of functions and derivatives for Newton Rapson
C solution of Hengstebeck-Geddes equations and Gillilands
C correlation.
C

```

```
C CODED BY: Jonathan Knight
C Dept Chem. Engng., MIT
C January 5, 1992
C
C Inputs are:
C  nc - I*4 number of components
C  NN - I*4 number of trays
C  RR - dp reflux ratio
C  phi - dp Underwood Equation constant
C  a  - dp array of separation factors
C  xb - dp array of bottom compositions
C  xd - dp array of overhead compositions (first value is an old value)
C  C1 - dp initial guess of H-G constant
C
C Outputs:
C  xd(1), C1
C
C Notes:
C  Jan 4: 25 set as upper limit on number components in mixture.
C
      subroutine hg_eval(a,C1,nc,nn,phi,rr,xb,xd,G1,G2,G1x,G1C,
      & G2x,G2C,k,xdr,xbr,ar,reference)
#include "equip_params.f"
C
C *** args
C
      integer*4 nc,nn,k,reference
      real*8 a(*),c1,phi,rr,xb(*),xd(*)
C      (functions and derivatives for newtons method)
      real*8 G1x, G1C, G2x, G2C
      real*8 G1, G2,ratio,xdr,xbr,ar

      real*8 xx,xmin,dgdr,drdx,drdc,
      & dgdx,dgdc
C
C *** locals
C
C      (equations whose roots are to be found.  see Diwekar., IECR, 1991, 713.)

      integer*4 i,j
C
C      compute G1 at old value

      G1 = 0.0
      do 1001 j=1,nc
        G1 = G1 + (a(j)/ar)**C1 * xb(j)
1001      continue
C
      G1 = G1*xdr/xbr - 1.0
C
C      compute d G1 / d C1 at old value

      G1C = 0.0
      do 1002 j=1,nc
        G1C = G1C + log(a(j)/ar) * (a(j)/ar)**C1 * xb(j)
1002      continue
C
      G1C = G1C*xdr/xbr

C      compute d G1 / d xd1 at old value
```



```

      G1x = (G1+1.0)/xdr

      rmin = 0.0

      do j=1,nc
        rmin = rmin + a(j)*xb(j)/(a(j)-phi)*
&          (a(j)/ar)**c1
&          * xdr / xbr
      end do
      rmin = rmin - 1.0

      xx = (rr-rmin)/(rr+1.0)

      if(xx .gt. 0.01 .and. xx .lt. 0.9) then
        g2 = (dble(nn)-c1)/(dble(nn)+1.0)
& - (0.545827-0.591422*xx + 0.002743/xx)
        dgdr = (-0.591422-0.002743/xx**2)/(rr+1.0)
      elseif(xx .lt. 0.01) then
        g2 = (dble(nn)-c1)/(dble(nn)+1.0)
& - (1.0-18.5715*xx)
        dgdr = -18.5715/(rr+1.0)
      else
        g2 = (dble(nn)-c1)/(dble(nn)+1.0)
& - (0.16595-0.16595*xx)
        dgdr = -0.16595/(rr+1.0)
      endif

      drdx = 0.0
      drdc = 0.0
      do j=1,nc
        drdx =
& drdx+a(j)*xb(j)/(a(j)-phi)*(a(j)/ar)**c1
& / xbr

        drdc =
& drdc + xdr/xbr*a(j)*xb(j)/(a(j)-phi)*
& (a(j)/ar)**c1 * log(a(j)/ar)
      end do

      g2x = dgdr*drdx
      g2c = -1.0/(dble(nn)+1.0) + dgdr*drdc

      return
      end

```

```

C
C Simple program using newtons method for computing Hengstebeck-Geddes
C constant.
C
C CODED BY: Jonathan Knight
C Dept Chem. Engng., MIT
C January 5, 1992
C
C Inputs are:

```

```
C   nc - I*4 number of components
C   a  - dp array of separation factors
C   xb - dp array of bottom compositions
C   xdl - dp overhead composition if light component
C   C1 - dp initial guess of H-G constant
C
C Outputs:
C   C1
C
C Notes:
C   Jan 4: 25 set as upper limit on number components in mixture.
C
C   subroutine get_c1(nc,a,xb,xdr,xbr,ar,C1)
#include "equip_params.f"
C
C *** args
C
C   integer*4 nc
C   real*8 a(*),c1,xb(*),xdl,xdr,xbr,ar
C
C
C *** locals
C
C   (equations whose roots are to be found. see Diwekar..., IECR, 1991, 713.)
C   real*8 G1
C
C   integer*4 i
C
C   (derivatives for newtons method)
C   real*8 G1C
C
C   differential change
C   real*8 dC
C
C   old values and error norm
C   real*8 Cold, err1, err2
C
C   initialize error norms to large values
C   err1 = 1.0E6
C   err2 = 1.0E6
C   i=1
1000 if(i .lt. 10 .and. (err1 .gt. 1.0E-6 .or.
&          err2 .gt. 1.0E-6)) then
C
C   make an evaluation of functions and derivatives
C   call c1_eval(nc,a,xb,xdr,xbr,ar,C1,G1,G1C)
C
C   save old values of xdl and C1 (only to compute error norm)
C   Cold = C1
C
C   compute differential changes
C   dC = -G1/G1C
C
C   update values
C   C1 = C1 + dC
C
C   compute error norms
C   err1 = sqrt(G1**2)
C   err2 = sqrt((C1-Cold)**2)
C
C   print out values
```

```
C      program control
C      i=i+1
C      goto 1000
C      endif

C      return
C      end

C
C Returns values of function and derivative for Newton Rapson
C solution of Hengstebeck-Geddes equation.
C
C CODED BY: Jonathan Knight
C Dept Chem. Engng., MIT
C January 5, 1992
C
C Inputs are:
C   nc - I*4 number of components
C   a  - dp array of separation factors
C   xb - dp array of bottom compositions
C   xd1 - dp distillate composition value
C   C1 - dp initial guess of H-G constant
C
C Outputs:
C   C1
C
C Notes:
C   Jan 4: 25 set as upper limit on number components in mixture.
C
C      subroutine c1_eval(nc,a,xb,xdr,xbr,ar,C1,G1,G1C)
#include "equip_params.f"
C
C *** args
C
C      integer*4 nc
C      real*8 a(*),c1,xb(*),xdr,xbr,ar
C      (functions and derivatives for newtons method)
C      real*8 G1C
C      real*8 G1

C
C *** locals
C
C      (equations whose roots are to be found.  see Diwekar..., IECR, 1991, 713.)

C      integer*4 i,j

C      compute G1 at old value
C      G1 = 0.0

C      do 1001 j=1,nc
C          G1 = G1 + (a(j)/ar)**C1 * xb(j)
1001      continue
C
C      G1 = G1*xdr/xbr - 1.0

C
C      compute d G1 / d C1 at old value

C      G1C = 0.0
```

```

        do 1002 j=1,nc
            G1C = G1C + log(a(j)/ar) * (a(j)/ar)**C1 * xb(j)
1002    continue
C
        G1C = G1C*xdr/xbr
        return
        end

C subroutine to compute distillate compositions, xd(i), i>1
C
C coded by : jonathan knight
C dept. che, MIT
C Jan 5, 1992
C
C Inputs are
C   C1 - dp hg constant
C   nc - i*4 number of components
C   xb - dp array of still pot boiling points
C   dxbl - change in bottom still pot composition of comp. 1
C   xd1 - distillate composition of comp. 1
C   a - separation factors
C
C Outputs are
C   xb(i), i>1.

        subroutine get_xd(a,C1,nc,xb,xd,xdr,xbr,
            & ar,reference)
#include "equip_params.f"

C
C *** args
C
        integer*4 nc,reference
        real*8 C1, xb(*), a(*), xd(*),ar,xbr,xdr

C
C *** locals
C
        integer*4 i
        real*8 sum

C
        sum = 0.0
        do i=1,nc
            if(i .ne. reference) then
                xd(i) = (a(i)/ar)**C1 * xdr / xbr * xb(i)
                sum = sum + xd(i)
            endif
        end do

C
        do i=1,nc
            if(i .ne. reference) then
                xd(i) = xd(i)/sum * (1.0-xdr)
            endif
        end do

C
        return
        end

C
C Simple program using newtons method for computing Underwood constant.
```

```
C
C CODED BY: Jonathan Knight
C Dept Chem. Engng., MIT
C January 5, 1992
C
C Inputs are:
C   nc - I*4 number of components
C   a  - dp array of separation factors
C   xb - dp array of bottom compositions
C
C Outputs:
C   phi - dp underwoods constant
C
C Notes:
C   Jan 4: 25 set as upper limit on number components in mixture.
C
C       subroutine get_phi(a,xb,nc,reference,phi)
#include "equip_params.f"
C
C *** args
C
C       integer*4 nc,reference
C       real*8 a(*),xb(*),phi
C
C *** locals
C
C       Underwoods equations
C       real*8 U1
C
C       integer*4 i
C
C       (derivative for newtons method)
C       real*8 DU1
C
C       differential change
C       real*8 dp
C
C       old values and error norm
C       real*8 pold, err1, err2
C
C       initialize phi
C       if(reference .lt. nc) then
C           phi = ( a(1) + a(2) ) / 2.0
C       endif
C
C       initialize error norms to large values
C       err1 = 1.0E6
C       err2 = 1.0E6
C       i=1
1000 if(i .lt. 10 .and. (err1 .gt. 1.0E-6 .or.
C           &                err2 .gt. 1.0E-6)) then
C
C           make an evaluation of functions and derivatives
C           call phi_eval(nc,a,xb,phi,U1,DU1)
C
C           save old value of phi (only to compute error norm)
C           pold = phi
C
C           compute differential changes
```

```
        dp = -U1/DU1

C      update values
        phi = phi + dp

C      compute error norms
        err1 = sqrt(U1**2)
        err2 = sqrt((phi-pold)**2)

C      program control
        i=i+1
        goto 1000
    endif

        return
    end

C
C Returns minimum reflux ratio computed from underwood equation
C
C CODED BY: Jonathan Knight
C Dept Chem. Engng., MIT
C January 5, 1992
C
C Inputs are:
C   nc - I*4 number of components
C   a  - dp array of separation factors
C   xd - dp array of distillate compositions
C   phi - dp underwoods constant
C
C Outputs:
C   rmin
C
C Notes:
C   Jan 4: 25 set as upper limit on number components in mixture.
C
        subroutine get_rmin(nc,a,xd,phi,rmin)
#include "equip_params.f"
C
C *** args
C
        integer*4 nc
        real*8 a(*),xd(*),phi
C      (functions and derivatives for newtons method)
        real*8 rmin

C
C *** locals
C
        integer*4 i,j

C      compute rmin
        rmin = 0.0
        do j=1,nc
            rmin = rmin + a(j)*xd(j)/(a(j)-phi)
        end do
        rmin = rmin - 1.0

C
        return
    end
```

```

C-----
C GET_TRAYS (FORTRAN 77)
C subroutine to compute the number of trays
C
C coded by and property of jonathan knight
C dept. chemical engineering
C MIT
C
C created Feb 15,1993
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      subroutine get_trays(rr,rmin,c1,nn)
#include "equip_params.f"
C
C ** ARGVS
C
      integer*4 nn
      real*8 c1,rr,rmin
C
C ** LOCALS
C
      real*8 g,psi

      psi = (rr-rmin)/(rr+1.0)
      g = 1.0 -
      & exp((1+54.4*psi) /
      & (11.0+117.2*psi) * (psi-1.0) / psi**0.5)

      nn = idnint((c1+g)/(1.0-g)+0.5)
C
      return
      end

C
C Returns values of function and derivative for Newton Rapson
C solution of Underwood root.
C
C CODED BY: Jonathan Knight
C Dept Chem. Engng., MIT
C January 5, 1992
C
C Inputs are:
C nc - I*4 number of components
C a - dp array of separation factors
C xb - dp array of bottom compositions
C phi - dp underwoods constant
C U1 - equation residual
C DU1 - derivative
C
C Outputs:
C phi
C
C Notes:
C Jan 4: 25 set as upper limit on number components in mixture.
C
      subroutine phi_eval(nc,a,xb,phi,U1,DU1)
#include "equip_params.f"
C

```

```
C *** args
C
  integer*4 nc
  real*8 a(*),xb(*),phi
C (functions and derivatives for newtons method)
  real*8 U1
  real*8 DU1

C
C *** locals
C
  integer*4 i,j
C
C   compute U1 at old value
  U1 = 0.0
  do j=1,nc
    U1 = U1 + a(j) * xb(j) / ( a(j) - phi )
  end do
C
C   compute d U1 / d phi at old value

  DU1 = 0.0
  do 1002 j=1,nc
    DU1 = DU1 + a(j) * xb(j) / ( a(j) - phi )**2
1002  continue
C
  return
  end
```

A5.2. Include File

```
C maximum number of components used in distillation
  integer*4 p_max_components
  parameter(p_max_components=10)

C max number of major iterations
  integer*4 p_loop_limit
  parameter(p_loop_limit=10)

C rr to rmin factor
  real*8 p_reflux_factor
  parameter(p_reflux_factor=1.32)

C ratio of boilup to feed size
  real*8 p_boilup_factor
  parameter(p_boilup_factor=0.333)

C default superficial velocity (taken from Luyden et al., 1991 IEC) (ft/s)
  real*8 p_vm_default
  parameter(p_vm_default=5.22)

C default diameter factor (same source as "p_vm_default")
  real*8 p_dfactor_default
  parameter(p_dfactor_default=0.02)

C default feed enthalpy (btu/lb-mole)
  real*8 p_default_feed_hv
  parameter(p_default_feed_hv=13700.0)
```



```
C default feed enthalpy (btu/lb)
  real*8 p_default_steam_hv
  parameter(p_default_steam_hv=915.5)

C default U DT value for reboiler
  real*8 p_default_rdt
  parameter(p_default_rudt=2500.0)

C default U DT value for condenser
  real*8 p_default_cudt
  parameter(p_default_cudt=2000.0)

C default material of construction factor for condenser
  real*8 p_default_rfex
  parameter(p_default_rfex=1.0)

C default material of construction factor for condenser
  real*8 p_default_cfex
  parameter(p_default_cfex=1.0)

C default time horizon for project (hours)
  real*8 p_default_time_horizon
  parameter(p_default_time_horizon=8500.0)

C default unit steam cost (i.e., $$ per lb)
  real*8 p_default_steam_cost
  parameter(p_default_steam_cost=0.005)

C heat exchanger time cost factor
  real*8 p_hex_factor
  parameter(p_hex_factor=5.0)

C default material density for column shell (lb/ft3)
  real*8 p_default_mat_density
  parameter(p_default_mat_density=500.0)

C default shell wall thickness
  real*8 p_default_wall_thick
  parameter(p_default_wall_thick=0.02083)

C default tray length
  real*8 p_default_tray_len
  parameter(p_default_tray_len=2.0)

C pi
  real*8 pi
  parameter(pi=3.141593)

C default tray material cost factors
  real*8 p_default_ftma,p_default_ftmb
  parameter(p_default_ftma=1.0)
  parameter(p_default_ftmb=0.0)

C default shell material factor
  real*8 p_default_fm
  parameter(p_default_fm=1.0)

C column "cost over time" factor
  real*8 p_col_factor
  parameter(p_col_factor=5.0)
```

Appendix 6: Code and Production Results for Reaction Path Network Algorithm

A6.1. Comments

This appendix provides a listing of the reaction path synthesis code. Computational output from the production runs in chapter eight is included.

A6.2. Computational Code

A6.2.1. Makefile

```
# makefile for retrosynthetic analysis code
#
OBJFILES1 = \
compare_mol.o compare_mol2.o local_ra.o local_ra_phase1.o \
prune_driver.o ra_driver.o local_ra_phase2.o get_byprods.o \
get_rate_data.o get_rxn_data.o update_byprod_list.o \
update_ast_list.o evaluate_rxn_path.o cchem.o
OBJFILES2 = \
compare_mol.f compare_mol2.f local_ra.f local_ra_phase1.f \
prune_driver.f ra_driver.f local_ra_phase2.f get_byprods.f \
get_rate_data.f get_rxn_data.f update_byprod_list.f \
update_ast_list.f evaluate_rxn_path.f cchem.o
#
compare_mol.o : compare_mol.f params.f
    f77 -o compare_mol.o -cpp -c compare_mol.f;
compare_mol2.o : compare_mol2.f params.f
    f77 -o compare_mol2.o -cpp -c compare_mol2.f;
local_ra.o : local_ra.f params.f
    f77 -o local_ra.o -cpp -c local_ra.f;
local_ra_phase1.o : local_ra_phase1.f params.f
    f77 -o local_ra_phase1.o -cpp -c local_ra_phase1.f;
local_ra_phase2.o : local_ra_phase2.f params.f
    f77 -o local_ra_phase2.o -cpp -c local_ra_phase2.f;
prune_driver.o : prune_driver.f params.f
    f77 -o prune_driver.o -cpp -c prune_driver.f;
ra_driver.o : ra_driver.f params.f
    f77 -o ra_driver.o -cpp -c ra_driver.f;
get_byprods.o : get_byprods.f params.f
    f77 -o get_byprods.o -cpp -c get_byprods.f;
get_rate_data.o : get_rate_data.f params.f
    f77 -o get_rate_data.o -cpp -c get_rate_data.f;
get_rxn_data.o : get_rxn_data.f params.f
    f77 -o get_rxn_data.o -cpp -c get_rxn_data.f;
update_byprod_list.o : update_byprod_list.f params.f
    f77 -o update_byprod_list.o -cpp -c update_byprod_list.f;
update_ast_list.o : update_ast_list.f params.f
    f77 -o update_ast_list.o -cpp -c update_ast_list.f;
evaluate_rxn_path.o : evaluate_rxn_path.f params.f
    f77 -o evaluate_rxn_path.o -cpp -c evaluate_rxn_path.f;
cchem.o: cchem.c
    cc -o cchem.o -cpp -c cchem.c;
#
# --- executable generation
#
# version1
prune: ${OBJFILES1}
    f77 -o prune ${OBJFILES1} ;
#
prune2: ${OBJFILES2}
    f77 -o prune -cpp ${OBJFILES2};
#
```

A6.2.2. Prune_Driver.f

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C PRUNE_DRIVER.F (FORTRAN 77)
C program which implements the following method for variable size
C branch and bound
C
C     (i). attempt to fathom current node
C     (ii). if (i) fails, then branch
C     (iii). if (i,ii) fail, then stop
C
C     CODED BY AND PROPERTY OF: JONATHAN P. KNIGHT
C     DEPT. CHEMICAL ENGINEERING
C     MIT
C     (created 01/26/1992)
C     Jan 29, 1992 .. introduced parameterization
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
program prune_driver

#include "params.f"

C *** MAJOR VARIABLES

C --> CURRENT POSITION IN STACK
integer*4 current_level,
& curr_node(p_max_level)

C --> NUMBER OF NODES AT EACH LEVEL OF STACK
integer*4 num_nodes(p_max_level)

C --> BRANCH AND FATHOM DECISION VARIABLES
integer*4 branch, fathom

C --> PROGRAM TERMINATION VARIABLE (i.e., TERMINATE IF
C NO PATH CAN BE FOUND)
integer*4 path_found

C --> SET OF CHEMICAL REACTION DESCRIPTORS AND THE TOTAL
C NUMBER OF REACTIONS. "grxn_add" IS THE SET OF GROUPS
C TO BE ADDED DURING A CHEMICAL REACTION. "grxn_remove"
C IS THE SET OF GROUPS REMOVED DURING A REACTION. BOTH INCLUDE
C HYDROGEN. THE GROUPS ARE REPRESENTED BY USER-DEFINED
C INTEGER INDICES, WITH HYDROGEN=1.† THE ARRAY "num_rxn"
C CONSISTS OF A FIRST ELEMENT WHICH DEFINES THE REACTION GROUP
C AND SUCCEEDING ELEMENTS WHICH DEFINE THE STARTING LOCATION OF
C THE REACTION GROUP IN THE REACTION LIST.
integer*4 grxn_add(p_max_num_rxn,p_num_site),
& grxn_remove(p_max_num_rxn,p_num_site),
& num_rxn(p_max_num_rxn)

C --> PRODUCT MOLECULE WHICH IS PRECURSOR TO NO OTHER MOLECULE IN
C THE ANALYSIS
integer*4 gproduct(p_num_site)

C --> PRODUCT MOLECULE (i.e., IDENTITY OF PARENT NODE) USED IN THE
C GENERATION OF PRECURSORS. *** MAY BE PRECURSOR TO OTHER
C PRODUCTS. THIS VARIABLE IS JUST THE MOLECULAR IDENTITY OF A
C NODE WHICH IS BEING ANALYZED RETROSYNTHETICALLY.
integer*4 product(p_num_site)
```

```
C --> GROWING LIST OF NODE MOLECULES GENERATED DURING
C RETROSYNTHETIC ANALYSIS AND THE TOTAL NUMBER OF MOLECULES
C GENERATED.
  integer*4 gnode_list(p_max_molecules,p_num_site),
  &         gnode_card

C --> GROWING LIST OF BYPRODUCT MOLECULES GENERATED
C DURING RETROSYNTHETIC ANALYSIS AND THE TOTAL NUMBER
C OF MOLECULES GENERATED.
  integer*4 byprod_list(p_max_molecules,p_num_site),
  &         byprod_card

C --> IDENTITIES OF MOLECULES (i.e., POINTS INTO "gnode_list")
C GENERATED IN THE FATHOMING OF A PARENT NODE, AND THE
C TOTAL NUMBER OF MOLECULES IN THE LIST
  integer*4 child_node_stack(p_max_nodes_per_level),
  & num_child_nodes

C --> PRECURSORS ASSOCIATED WITH REACTIONS. I.E., POINTS INTO
C "child_node_stack". "num_precursor_list" IS THE NUMBER OF
C PRECURSORS FOR THE REACTION.
  integer*4 rxn_node_ptr(p_max_num_rxn,
  &                     p_max_precursor_per_rxn),
  & num_precursor_list(p_max_num_rxn)

C --> STACK VERSION OF "rxn_node_ptr"
  integer*4 rxn_ptr_stack(p_max_level,
  & p_max_num_rxn,p_max_precursor_per_rxn),
  & rxn_ptr_list(p_max_level,p_max_num_rxn)

C --> MATRIX OF REACTION DATA FOR EACH REACTION IN A LEVEL
  real*8 rxn_data(p_max_num_rxn,p_max_precursor_per_rxn,
  &              p_max_rxn_data)

C --> STACK VERSION OF "rxn_data"
  real*8 rxn_data_stack(p_max_level,
  & p_max_num_rxn,p_max_precursor_per_rxn,p_max_rxn_data)

C --> BYPRODUCTS ASSOCIATED WITH REACTIONS. I.E., POINTS INTO
C "child_node_stack". "num_byrod_list" IS THE NUMBER OF
C PRECURSORS FOR THE REACTION AND PRECURSOR PAIR.
  integer*4 rxn_byprod(p_max_num_rxn,
  & p_max_precursor_per_rxn,p_max_precursor_per_rxn),
  & num_byprod_list(p_max_num_rxn,p_max_precursor_per_rxn)

C --> STACK VERSION OF "rxn_byprod"
  integer*4 byprod_stack(p_max_level,p_max_num_rxn,
  & p_max_precursor_per_rxn,p_max_precursor_per_rxn),
  & byprod_stack_list(p_max_level,p_max_num_rxn,
  & p_max_precursor_per_rxn)

C --> BYPRODUCT YIELDS. I.E., POINTS INTO
C "child_node_stack". "num_byrod_list" IS THE NUMBER OF
C PRECURSORS FOR THE REACTION AND PRECURSOR PAIR.
  real*8 byprod_yields(p_max_num_rxn,
  & p_max_precursor_per_rxn,p_max_precursor_per_rxn)

C --> STACK VERSION OF "byprod_yields"
  real*8 yield_stack(p_max_level,p_max_num_rxn,
  & p_max_precursor_per_rxn,p_max_precursor_per_rxn)
```

```

C --> STACK OF NODES WHICH IS FULL UP TO THE CURRENT LEVEL (DEEPER
C     LEVELS ARE AUTOMATICALLY ZEROED OUT).  EACH ELEMENT POINTS
C     TO A MOLECULE IN "gnode_list"
      integer*4 node_stack(p_max_level,p_max_nodes_per_level)

C --> IDENTITY OF LEVEL AND NODE WHICH ARE TO EITHER BE (1) FATHOMED
C     OR (2) BRANCHED AWAY FROM.
      integer*4 parent_level, parent_node

C --> LIST OF MOLECULE STRUCTURES (i.e., NODES) WHICH HAVE BEEN COMPLETELY
C     FATHOMED, AND THE CARDINALITY OF THE ARRAY
      integer*4 fathomed_list(p_max_molecules),num_fathomed

C --> (=1) IF THE CURRENT PARENT NODE HAS ALREADY BEEN FATHOMED, 0
C     OTHERWISE
      integer*4 completely_fathomed

C --> CURRENT INDEX IN "gnode_list" TO PARENT MOLECULE
      integer*4 parent_idx

C --> CURRENT TERMINAL NODE WHICH MAY OR MAY NOT BE A COMPLETELY
C     FATHOMED ENTRY, A FLAG WHICH DETERMINES THE CASE TO BE, AND
C     A FLAG WHICH STOPS THE SEARCH LOOP USED IN THE DETERMINATION
      integer*4 fathomed_entry_found, fathomed_entry, stop_search

C --> LIST OF FEASIBLE AGGREGATE STATE TASK (AST) SEQUENCES (INTEGER
C     AND REAL DATA), AND CARDINALITY
      real*8 ast_listr(p_max_ast,p_num_ast_datar)
      integer*4 ast_listi(p_max_ast,p_num_ast_datai),
&          ast_card

C --> NUMBER OF DISTINCT REACTION GROUPS.  THIS IS ESSENTIALLY AN
C     ELEMENT OUT OF THE ARRAY "num_rxn"
      integer*4 total_num_rxn

C *** AUXILIARY VARIABLES

C --> MISCELLANEOUS INDICES
C     NOTE: "scratch_idx" IS USED TO GET AN INDEX WHICH
C           STARTS AT "1" AT THE BEGINNING OF A DO LOOP
      integer*4 i,j,k,m,n,scratch_idx
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C PDL:
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      current_level = 0
      parent_level = 0
      parent_node = 0
      num_fathomed = 0
      ast_card = 0

C ** READ IN GLOBAL PRODUCT
      call get_product(gproduct)

C ** READ IN CHEMICAL REACTIONS AND ASSIGN "total_num_rxn"
      call get_rxns(grxn_add,grxn_remove,num_rxn)
      total_num_rxn = num_rxn(p_tot_num_rxn)

C *** INITIALIZE PRODUCT NODE AS THE TARGET MOLECULE
      do 9000 j=1,p_num_site

```

```

        product(j) = gproduct(j)
        gnode_list(1,j) = gproduct(j)
9000  continue
        gnode_card = 1

C *** INITIALIZE BYPRODUCT CARDINALITY
        byprod_card = 0
C
        do 1000 j=1,p_max_it
C
C ** INITIALIZE SOLUTION STATUS (NEW NODE NOT FOUND) AND SET THE
C CURRENT NODE TO BE THE PARENT
        path_found = 0
C
        parent_level = current_level
        parent_node = curr_node(parent_level)

C ** CRITICAL: GENERATE CHILD NODES AND DECIDE WHETHER TO FATHOM
C (I) SELECT THE PRODUCT MOLECULE
C (II) CHECK TO SEE IF THE PRODUCT MOLECULE HAS BEEN COMPLETELY
C FATHOMED
C (IIA) IF THE MOLECULE HAS BEEN FATHOMED, THEN BRANCH.
C (IIB) OTHERWISE, CHECK GENERATE CHILDREN NODES AND IF THERE ARE
C ANY, THEN FATHOM (OTHERWISE BRANCH)
C
        if (parent_level .gt. 0) then
            parent_idx = node_stack(parent_level,parent_node)
C
            do 5000 k=1,p_num_site
                product(k) = gnode_list(parent_idx,k)
5000  continue
            endif
C
            completely_fathomed = 0
C
            call check_node(parent_idx,fathomed_list,
                & num_fathomed,completely_fathomed)

C ** THIS IF STATEMENT IS IN EFFECT UNTIL THE GENERAL BRANCHING
C RATIONALE IS DEVELOPED
        if(completely_fathomed .eq. 1 .or. parent_level .ge.
            & p_max_level) then
            fathom = 0
            branch = 1
        else
C
            call ra_driver(grxn_add,grxn_remove,total_num_rxn,
                & num_rxn(p_rxn_set_size_start),product,
                & gnode_list,gnode_card,byprod_list,
                & byprod_card,byprod_yields,
                & child_node_stack,num_child_nodes,
                & rxn_node_ptr,num_precursor_list,
                & rxn_byprod,num_byprod_list,rxn_data)
C
C *** THE OUTPUT FROM RADRIVER SHOULD CONTAIN ONLY FEASIBLE AGGREGATE STATE-
C TASKS, WHICH CAN BE DUMPED OUT. THEN A SCREENING OBJECTIVE FUNCTION
C AT THIS LEVEL SHOULD BE APPLIED, AND THE TASKS WHICH SATISFY IT SHOULD
C BE ADDED TO AGGREGATE STATE-TASK LIST. AS INFEASIBLE REACTIONS ARE
C IDENTIFIED, THEIR POINTER SHOULD BE RESET TO THE NULL POINTER, THEN THE
C UPDATE CAN PROCEED ON THE BASIS IDENTIFYING RXNS WITH NON-NULL POINTERS

```

```

C
    if(parent_level .gt. 0) then
        call evaluate_rxn_path(parent_level,total_num_rxn,
&         curr_node,rxn_ptr_stack,rxn_ptr_list,
&         rxn_data_stack,rxn_node_ptr,
&         num_precursor_list,rxn_data,
&         child_node_stack,num_child_nodes)
    endif

    call update_ast_list
&         (total_num_rxn,product,gnode_list,
&         gnode_card,
&         byprod_list,byprod_card,
&         byprod_yields,
&         child_node_stack,num_child_nodes,
&         rxn_node_ptr,num_precursor_list,
&         rxn_byprod,num_byprod_list,rxn_data,
&         ast_listi,ast_listr,ast_card)

    if(num_child_nodes .gt. 0) then
        fathom = 1
        branch = 0
    endif
    if(num_child_nodes .eq. 0) then
        fathom = 0
        branch = 1
    endif
endif
endif

C ** IF NODE IS BEING FATHOMED, THEN UPDATE THE MOLECULE AND
C REACTION STACK.
C     (I) INCREMENT CURRENT LEVEL
C     (II) READ IN THE CURRENT MOLECULE STACK
C     (III) READ IN THE REACTION POINTER STACK (THIS STACK
C           IS MORE GENERAL THAN II, AND IN FACT POINTS INTO
C           II)
C     (IV) SINCE A NEW LEVEL IS BEING FATHOMED, SET NODE AT
C           THAT LEVEL TO 1
C     (V) INDICATE THAT A PATH HAS BEEN FOUND
if (fathom .eq. 1) then

    call update_level(current_level)
    num_nodes(current_level) = num_child_nodes
    do 5001 k=1,num_child_nodes
        node_stack(current_level,k) = child_node_stack(k)
5001    continue

    do 5002 k=1,total_num_rxn
        rxn_ptr_list(current_level,k) =
&         num_precursor_list(k)
        do 5003 m=1,num_precursor_list(k)

C ** PRECURSOR POINTERS
&         rxn_ptr_stack(current_level,k,m) =
&         rxn_node_ptr(k,m)

C ** REACTION DATA STACK
        do 5999 n=1,p_max_rxn_data
&         rxn_data_stack(current_level,k,m,n) =
&         rxn_data(k,m,n)

```



```

5999          continue

C ** BYPRODUCT POINTERS
      byprod_stack_list(current_level,k,m) =
&          num_byprod_list(k,m)
      do 6000 n=1,num_byprod_list(k,m)
&          byprod_stack(current_level,k,m,n)
&          = rxn_byprod(k,m,n)
&          yield_stack(current_level,k,m,n)=
&          byprod_yields(k,m,n)
6000          continue
5003          continue
5002          continue

      curr_node(current_level) = 1
      path_found = 1

      endif
      if (branch .eq. 1) then
        do 2000 k = current_level,1,-1
          if (path_found .eq. 0 .and. curr_node(k) .lt.
&          num_nodes(k) ) then

            path_found = 1
            current_level = k
            call update_node(curr_node,current_level)
          endif

          if (path_found .eq. 0 .and. curr_node(k) .eq.
&          num_nodes(k) .or. current_level .gt.
&          p_max_level) then

            stop_search = num_fathomed
            fathomed_entry_found = 0
            fathomed_entry = node_stack(k,curr_node(k))
            do 5007 m=1,stop_search
              if(fathomed_entry.eq.fathomed_list(m))then
                fathomed_entry_found = 1
                stop_search = m
              endif
            endif
5007          continue
            if (fathomed_entry_found .eq. 0) then
              num_fathomed = num_fathomed+1
              fathomed_list(num_fathomed) =
&              node_stack(k,curr_node(k))
            endif
          endif
        endif

        if (path_found .eq. 0 .and. curr_node(k) .eq.
&        num_nodes(k) ) curr_node(k) = 0

2000          continue

          if (path_found .eq. 0) then
            call ast_out(ast_card,byprod_card,gnode_card,
&            ast_listi,ast_l:str,gnode_list,byprod_list)
            stop
          endif
        endif
      endif

C ***
1000          continue

```

```
        stop
        end

C-----
C subroutine to read in the chemical reactions and pass them to the
C driver routine. ***IT IS IMPORTANT TO RECOGNIZE THAT THE FIRST REACTION
C MUST BE "1".***

        subroutine get_rxns(grxn_add,grxn_remove,num_rxn)
#include "params.f"

C
C *** args
C
        integer*4 grxn_add(p_max_num_rxn,*),
        & grxn_remove(p_max_num_rxn,*),num_rxn(*)

C
C ***** locals
C
C --> INPUT FILE CONTAINING REACTION SPECIFICATIONS
        character*80 infile

C --> TOTAL NUMBER OF GROUPS UNDER CONSIDERATION
        integer*4 num_groups

C --> MISC. INDICES
        integer*4 i,j
C
        write(*,*) '---ENTER REACTION INPUT FILE NAME'
        read(*,'(A80)') infile
        open(unit = 12,file = infile ,status = 'old')

C ** READ IN THE NUMBER OF REACTION SETS AND THE STARTING POINT OF
C EACH GROUP IN THE REACTION LIST.
C PER SET
        read(12,*) num_rxn(p_tot_num_rxn),num_groups,
        & (num_rxn(j),j=
        & p_rxn_set_size_start,
        & num_groups+p_tot_num_rxn)

C ** SOMETIMES THE ALGORITHMS THAT UTILIZE THE GROUPING DATA LOOK
C ONE ELEMENT TOO FAR. SET THE ADDITIONAL ELEMENT TO THE LAST
C RXN PLUS ONE
        num_rxn(num_groups+p_tot_num_rxn+1) =
        & num_rxn(p_tot_num_rxn) + 1

C ** READ IN THE INDIVIDUAL REACTIONS, WHICH ARE GROUPED ACCORDING TO
C THE "num_rxn()" SET. NOTE THEY MUST BE INPUT IN THE CORRECT
C ORDER
        do 1000 i=1,num_rxn(p_tot_num_rxn)
            read(12,*) (grxn_add(i,j),j=1,p_num_site),
            & (grxn_remove(i,j),j=1,p_num_site)
1000 continue
        close(12)

C
        return
        end
```



```

C TO SEE IF MAXIMUM LEVEL HAS BEEN SUCCEEDED; AND IF SO, TERMINATES
C WITH A STATUS MESSAGE
  subroutine update_level(current_level)
#include "params.f"
  integer*4 current_level
C
  if(current_level .ge. p_max_level) then
    write(*,*) '***MAX LEVEL EXCEEDED, PROGRAM TERMINATED'
    stop
  else
    current_level = current_level+1
  endif
C
  return
  end

C -----
C CODE FOR UPDATING THE CURRENT NODE (I.E., INCREMENT BY 1). CHECKS
C TO SEE IF MAXIMUM LEVEL HAS BEEN SUCCEEDED; AND IF SO, TERMINATES
C WITH A STATUS MESSAGE
  subroutine update_node(curr_node,current_level)
#include "params.f"
  integer*4 curr_node(*),current_level
C
  if(curr_node(current_level) .ge. p_max_nodes_per_level)
& then
    write(*,*) '***MAX NODE EXCEEDED, PROGRAM TERMINATED'
    write(*,*) '  TERMINATED AT LEVEL = ',current_level
    stop
  else
    curr_node(current_level) = curr_node(current_level)+1
  endif
C
  return
  end

C -----
C CODE FOR PRINTING OUT RESULTS OF AGGREGATE STATE TASKS
  subroutine ast_out(ast_card,byprod_card,gnode_card,
& ast_listi,ast_listr,gnode_list,byprod_list)
#include "params.f"
C
C *** ARGS
  real*8 ast_listr(p_max_ast,*)
  integer*4 ast_listi(p_max_ast,*),
& ast_card,
& gnode_card,
& byprod_card
  integer*4 gnode_list(p_max_molecules,* )
  integer*4 byprod_list(p_max_molecules,*)

C
C *** LOCALS
C
C --> INDEX TO PRECURSOR AND BYPRODUCT FROM THE RESPECTIVE
C GLOBAL ARRAYS; UTILIZED IN OUTPUT
  integer*4 precursor_idx,byprod_idx

C --> INDEX INTO "ast_datar()" FOR THE YIELD INDEX
  integer*4 yield_idx

```

```

C --> INDEX INTO "ast_datai()" FOR THE RELATIVE INDEX TO THE
C PARENT WITHIN THE BYPRODUCT INDICES GIVEN IN "ast_datai()"
integer*4 product_idx

C *** AUXILIARY VARIABLES

C --> MISC INDICES
C NOTE: "scratch_idx" IS USED TO GET AN INDEX WHICH
C STARTS AT "1" AT THE BEGINNING OF A DO LOOP
integer*4 ii,jj,kk,scratch_idx

C --> OPEN OUTPUT FILE FOR RAW MOLECULAR STRUCTURES
open(unit = 15,file = 'rawmol.out' ,status = 'unknown')

do ii=1,ast_card

C ** PRINT OUT HEADER
write(*,'(1x,A24)') '-----'
write(*,'(1x,A24,I4)') '--- CHEMICAL CONVERSION ',ii
write(*,'(1x,A24)') '-----'

C ** COMPUTE THE INDICES REQUIRED TO RETRIEVE THE
C PRECURSOR AND TO IDENTIFY THE PRODUCT
precursor_idx=ast_listi(ii,p_ast_pre_id)
product_idx=ast_listi(ii,p_ast_par_locate)

C ** WRITE OUT THE PRECURSOR STRUCTURE
write(*,'(1x,A15,3x,6I4)') ' PRECURSOR: ',
& (gnode_list(precursor_idx,jj),jj=1,6)

C ** HERE JUST WRITE OUT PRECURSOR STRUCTURE SO WE CAN COUNT THEM.
write(15,'(3x,6I4)') (gnode_list(precursor_idx,jj),jj=1,6)

C ** FOR EACH BYPRODUCT, WRITE OUT THE IDENTITY
C AND YIELD. IF THE BYPRODUCT IS ALSO THE MAIN
C PRODUCT, INDICATE THIS WITH AN ASTERISK AND WRITE
C OUT THE RELATIVE RATE AS WELL.
do jj=p_ast_byprod_start,p_num_ast_datai
scratch_idx=jj-p_ast_byprod_start+1
byprod_idx=ast_listi(ii,jj)
yield_idx=jj-p_ast_byprod_start
if(scratch_idx.eq. product_idx) then
if(byprod_idx .ne. 0) then
write(*,
& '(1x,A15,3x,6I4,A11,3x,F7.5,A11,3x,F13.5)')
& ' *BYPRODUCT: ',
& (byprod_list(byprod_idx,kk),kk=1,6),
& ' YIELD =',
& ast_listr(ii,p_ast_yield_start+yield_idx),
& ' RATE =',
& ast_listr(ii,p_ast_rate_start)

C ** HERE JUST WRITE OUT BY-PRODUCT STRUCTURE SO WE CAN COUNT THEM.
write(15,'(3x,6I4)') (byprod_list(byprod_idx,kk),kk=1,6)
endif

else
if(byprod_idx .ne. 0) then
write(*,'(1x,A15,3x,6I4,A11,3x,F7.5)')

```

```

&          '   BYPRODUCT: ',
&          (byprod_list(byprod_idx,kk),kk=1,6),
&          '   YIELD = ',
&          ast_listr(ii,p_ast_yield_start+yield_idx)

C **      HERE JUST WRITE OUT BY-PRODUCT STRUCTURE SO WE CAN COUNT THEM.
          write(15,'(3x,6I4)') (byprod_list(byprod_idx,kk),kk=1,6)
          endif
        endif
      end do
      write(*,*) ' '
      write(*,*) ' '
    end do

C
C ** OUTPUT TO FILE FOR COMPARISON OF PRECURSORS WITH
C OTHER PRODUCT LINES
  open(unit = 14,file = 'molecules.out' ,status = 'unknown')
  write(14,*) gnode_card+byprod_card
  do ii=1,gnode_card
    write(14,'(1x,6I5)')
&    (gnode_list(ii,jj),jj=1,6)
  end do

  do ii=1,byprod_card
    write(14,'(1x,6I5)')
&    (byprod_list(ii,jj),jj=1,6)
  end do
  close(14)
  close(15)

  return
end

```

A6.2.3. Ra_Driver.f

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C RA_DRIVER.F (FORTRAN 77)

```

```

C Driver for retrosynthetic analysis -- applies a list of retro-
C reactions to a parent molecule. Generates a list of precursor
C molecules which is preprocessed based on relative rates to most
C rapidly establish bounds on pathway optimization. Relatedly,
C reaction pointers into the list of precursor molecules will take
C a NULL value if they are manifestly infeasible.

```

C

C

```

C          CODED BY AND PROPERTY OF JONATHAN KNIGHT

```

C

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

  subroutine ra_driver(grxn_add,grxn_remove,num_rxn,
&    rxn_group_idx,product,gnode_list,gnode_card,
&    byprod_list,byprod_card,
&    byprod_yields,
&    child_node_stack,num_child_nodes,
&    rxn_node_ptr,num_precursor_list,
&    rxn_byprod,num_byprod_list,rxn_data)

```

```

#include "params.f"

```

C

```

C *** args

```

```

C
C   external cchem01
C       integer*4 grxn_add(p_max_num_rxn,*),
C       & grxn_remove(p_max_num_rxn,*),
C       & num_rxn, rxn_group_idx(*)
C
C       integer*4 gnode_list(p_max_molecules,*),
C       & gnode_card
C
C       integer*4 byprod_list(p_max_molecules,*),
C       & byprod_card
C
C       integer*4 child_node_stack(*),
C       & num_child_nodes
C
C       integer*4 rxn_node_ptr(p_max_num_rxn,*),
C       & num_precursor_list(*)
C
C       integer*4 rxn_byprod(p_max_num_rxn,
C       & p_max_precursor_per_rxn,*),
C       & num_byprod_list(p_max_num_rxn,*)
C
C       real*8 rxn_data(p_max_num_rxn,
C       & p_max_precursor_per_rxn,*)
C
C       integer*4 product(p_num_site)
C
C       real*8 byprod_yields(p_max_num_rxn,
C       & p_max_precursor_per_rxn,*)
C
C *** locals
C
C --> LIST OF PRECURSORS FOR A GIVEN REACTION AND THE CARDINALITY
C       OF THE LIST
C       integer*4 node_indices(p_max_precursor_per_rxn),
C       & num_precursor
C
C --> PRECURSORS ASSOCIATED WITH REACTIONS. I.E., POINTS INTO
C       "child_node_stack". "num_precursor_list" IS THE NUMBER OF
C       PRECURSORS FOR THE REACTION.
C       integer*4
C       & rxn_node_indices(p_max_num_rxn,p_max_precursor_per_rxn)
C
C --> USED IN THE LOOP WHICH DIRECTS THE REACTION INDEX TO THE
C       COMPRESSED SET OF PRECURSORS. "idx_stop" MAKES THE SEARCHING
C       LOOP STOP SEARCHING WHEN THE POINTER IS IDENTIFIED
C       integer*4 idx_stop,idx_stop1,idx_stop2
C
C --> CURRENT PRECURSOR WHOSE IDENTITY IS USED TO GENERATE
C       POINTER FROM "rxn_node_ptr" into "child_node_stack"
C       integer*4 curr_mol_idx
C
C --> A NODE INDEX WITHIN THE REACTION INDICIES LIST
C       integer*4 node_idx
C
C --> FLAG INDICATING A POINTER FROM RXN LIST TO STRUCTURE LIST
C       HAS BEEN FOUND
C       integer*4 ptr_found

```

```

C --> SLICE OF DATA FOR A SINGLE REACTION. LOADED INTO
C   "rxn_data" ARRAY
      real*8 rxn_data_slice(p_max_precursor_per_rxn,
&                          p_max_rxn_data)

C --> AN INDIVIDUAL PRODUCT YIELD AND RATE FOR ONE OF THE REACTIONS
      real*8 pyield,prate

C --> POINTER INDICATING RESULT OF INDIVIDUAL STEP EVALUATION
      integer*4 eval_ptr

C --> AN INDEPENDENT INDEX WHICH DEFINES THE CURRENT REACTION GROUP
C   OF THE CURRENT REACTION BEING EVALUATED
      integer*4 group_count

C --> BEGINNING AND ENDING REACTIONS IN THE CURRENT REACTION GROUPING
      integer*4 rxn_group_start, rxn_group_stop

C misc indices
      integer*4 i,j,k,m,num_match,rxn_idx

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

C ** INITIALIZE UPPER AND LOWER BOUNDS ON REACTION GROUP
      group_count=1
      if(num_rxn .gt. 1) then
          rxn_group_start=rxn_group_idx(group_count)
          rxn_group_stop =rxn_group_idx(group_count+1)-1
      else
          rxn_group_start=1
          rxn_group_stop =1
      endif

      do 1000 i=1,num_rxn

C **      DYNAMICALLY UPDATE THE BOUNDS ON RXN GROUPINGS
          if(i .gt. rxn_group_stop) then
              group_count=group_count+1
              rxn_group_start=rxn_group_idx(group_count)
              rxn_group_stop =rxn_group_idx(group_count+1)-1
          endif

C **      GET THE PRECURSORS FOR THE CURRENT REACTION
          rxn_idx = i
          call local_ra(rxn_idx,rxn_group_start,rxn_group_stop,
& product,grxn_add,grxn_remove,
& node_indices,num_precursor,
& gnode_list,gnode_card,byprod_list,byprod_card,
& rxn_byprod,num_byprod_list,byprod_yields,
& rxn_data_slice)

C **      ADD REACTION DATA TO STACK ARRAY
          do 8001 j=1,num_precursor
              do 8002 k=1,p_max_rxn_data
                  rxn_data(i,j,k) = rxn_data_slice(j,k)
8002              continue
8001          continue

C **      ADD THE PRECURSOR INDICES TO THE CURRENT STACK LIST
          do 1003 j=1,num_precursor
              rxn_node_indices(i,j) = node_indices(j)

```



```
1003     continue
        num_precursor_list(i) = num_precursor
1000  continue

C ** COMPRESS THE INDICES INTO ONE UNIQUE LIST (PHASE 1)
C     (I) COMPILE THE LIST
C     (II) COMPRESS THE LIST
C     (III) PREPROCESS THE LIST
C !! THIS PART OF THE CODE COULD BE FURTHER OPTIMIZED
      k=0
      do 2001 i = 1,num_rxn
        do 2002 j = 1,num_precursor_list(i)
          node_idx = rxn_node_indices(i,j)

C ** RXNS THAT HAVE BEEN SCREENED OUT ARE AVOIDED BY SETTING
C THEIR POINTER VALUE TO A NULL VALUE, IN WHICH CASE THEIR
C IMPLIED PRECURSOR IS NEVER ADDED TO THE LIST.†I DONT SEE THE
C VALUE OF THE "IF" STATEMENT HERE SINCE NONE SHOULD HAVE HAD
C THEIR NULL VALUE SET.
          if(node_idx .ne. p_null_value) then
            k=k+1
            child_node_stack(k) = node_idx
          endif
2002     continue
2001  continue

C
      num_match = 0
      if ( k .gt. 1) then
        do 3000 i=1,k-num_match-1
          j=i+1
3001     if ( j .lt. k-num_match ) then
              if(child_node_stack(i) .eq.
& child_node_stack(j) ) then
                do 3002 m=j+1,k-num_match
                  child_node_stack(m-1) =
& child_node_stack(m)
3002         continue
                  num_match = num_match + 1
                  j=j-1
                endif
                j=j+1
                goto 3001
              endif
3000     continue
        endif

C ** COMPUTE THE NEW NUMBER OF CHILD NODES WHICH REFLECTS
C THE COMPRESSED CHILD NODE LIST.
      num_child_nodes = k - num_match

C
C ** BUILD THE REACTION POINTERS INTO CHILD_NODE_STACK. IF
C NO MOLECULE IS FOUND THEN THE POINTER IS SET TO A NULL
C VALUE.
      do 4000 i=1,num_rxn
        ptr_found = 0
        do 4001 j=1,num_precursor_list(i)
          curr_mol_idx = rxn_node_indices(i,j)
          idx_stop = num_child_nodes
```

```

do 4002 k=1,idx_stop
  if(child_node_stack(k) .eq. curr_mol_idx)
    & then
      pyield = rxn_data(i,j,p_rxn_yield)
      prate = rxn_data(i,j,p_rxn_rate)
      call eval_step(pyield,prate,eval_ptr)
      if(eval_ptr .ne. p_null_value) then
        rxn_node_ptr(i,j) = k
      else
        rxn_node_ptr(i,j) = p_null_value
      endif
      idx_stop = k
      ptr_found = 1
    endif
4002 continue
4001 continue
  if(ptr_found .eq. 0) rxn_node_ptr(i,j) = p_null_value
4000 continue

C ** SUPERCOMPRESSION OF NODE STACK (PHASE 2) TO ACCOUNT FOR
C THE FACT THAT NODES MAY HAVE BEEN REMOVED DUE TO
C PRUNING CRITERIA
  call phase2_pruning(num_child_nodes,child_node_stack,
  & num_rxn,num_precursor_list,rxn_node_ptr)

C ** THERE USED TO BE THE FOLLOWING PIECE OF CODE HERE, BUT IT
C APPEARS TO BE SPURIOUS LEFTOVER FROM WHEN PHASE 2 WAS
C CONVERTED INTO A SUBROUTINE:
C --> num_child_nodes = k - num_match

C ** THE FOLLOWING CODE IS THE BEGINNING OF THE COMPUTATIONAL
C CHEMISTRY ANALYSIS. EVERYTHING BEING CALLED IS WRITTEN
C IN C AND INVOLVES SYSTEM CALLS TO OTHER PROGRAMS. 8/25/93

C do iii = 1, gnode_card
C write(*,*) iii,'(F) ',(gnode_list(iii,jjj),jjj=1,6)
C end do

C ** IF ANY CHILD NODES ARE PRESENT, GO DO COMPUTATIONAL CHEMISTRY-BASED
C EVALUATION
C if(num_child_nodes .gt. 0) then
C call cchem01(num_child_nodes,child_node_stack,gnode_card,
C & gnode_list)
C endif

  return
end

C-----
C EVAL_STEP (FORTRAN 77)
C
C CODE TO EVALUATE AN INDIVIDUAL SYNTHETIC STEP
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
  subroutine eval_step(pyield,prate,eval_ptr)
#include "params.f"
  real*8 pyield,prate
  integer*4 eval_ptr
  eval_ptr=0
  if(pyield .lt. p_min_step_yield) eval_ptr = p_null_value

```

```

    if(prate .lt. p_min_step_rate) eval_ptr = p_null_value
    return
end

```

```

C-----
C PHASE2_PRUNING (FORTRAN 77)
C SUPERCOMPRESSION OF NODE STACK AND REORDERING OF RXN NODE POINTERS
C TO ENFORCE QUANTITATIVE PRUNING CRITERIA
C
C coded by jonathan knight
C dept. chemical engng.
C MIT
C
C created 2/6/93
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      subroutine phase2_pruning(num_child_nodes,child_node_stack,
        &          num_rxn,num_precursor_list,rxn_node_ptr)
#include "params.f"
C
C *** ARGS
C
      integer*4 child_node_stack(*),
        & num_child_nodes
C
      integer*4 rxn_node_ptr(p_max_num_rxn,*),
        & num_precursor_list(*)
C
      integer*4 num_rxn
C
C *** LOCALS
C
C --> USED IN THE LOOP WHICH DIRECTS THE REACTION INDEX TO THE
C COMPRESSED SET OF PRECURSORS. "idx_stop" MAKES THE SEARCHING
C LOOP STOP SEARCHING WHEN THE POINTER IS IDENTIFIED
      integer*4 idx_stop1,idx_stop2
C
C --> INDICATES A MOLECULE PRODUCING REACTION HAS/HAS NOT BEEN FOUND
      integer*4 hit,no_match
C
C misc indices
      integer*4 i,j,k,m,cnode,rnode,n

      no_match=0
      hit=1
      cnode=0
      do 7700 k=1,num_child_nodes-no_match

        if(hit .eq. 1) cnode=cnode+1

        hit=0
        do 7701 i=1,num_rxn
          do 7702 j=1,num_precursor_list(i)
            rnode=rxn_node_ptr(i,j)
            if(rnode .ne. p_null_value) then
              if(rnode .eq. cnode) then
                hit=1
              endif
            endif
          endif
        continue
7702

```

```

7701     continue
        if(hit .eq. 0) then

            do 7703 m=cnode+1,num_child_nodes-no_match
                child_node_stack(m-1) =
&         child_node_stack(m)

C ** MUST RECOMPUTE THE POINTER VALUES
        do 8800 i=1,num_rxn
            do 8801 n=1,num_precursor_list(i)
                rnode=rxn_node_ptr(i,n)
                if(rnode .eq. m) then
                    rnode=m-1
                    rxn_node_ptr(i,n)=rnode
                endif
8801         continue
8800     continue
7703     continue
            no_match = no_match + 1
        endif
7700 continue

        num_child_nodes=num_child_nodes-no_match

        return
        end

```

A6.2.4. Local_Ra.f

```

C-----
C LOCAL_RA.F (FORTRAN 77)
C subroutine controlling local retrosynthetic analysis
C presently, the code is to do the following
C     (1) generate precursors for a reaction... local_ra_phase1
C     (2) add the precursors to the global list, return
C         their index numbers and increment the cardinality
C         of the list... update_node_list
C
C     CODED BY AND PROPERTY OF JONATHAN KNIGHT
C     DEPT. CHEMICAL ENGINEERING
C     MIT
C     (created 02/03/93)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
    subroutine local_ra(rxn_idx,rxn_group_start,
&         rxn_group_stop,
&         product,grxn_add,grxn_remove,
&         gnode_indices,num_precursor,
&         gnode_list,gnode_card,byprod_list,byprod_card,
&         rxn_byprod,num_byprod_list,byprod_yields,
&         rxn_data_slice)
#include "params.f"
C
C *** args
C
    integer*4 product(*), grxn_add(p_max_num_rxn,*),
&         grxn_remove(p_max_num_rxn,*), gnode_card,
&         gnode_list(p_max_molecules,*),
&         gnode_indices(*),rxn_idx,rxn_group_start,

```

```

&          rxn_group_stop

integer*4 byprod_list(p_max_molecules,p_num_site),
&          byprod_card

integer*4 rxn_byprod(p_max_num_rxn,
&          p_max_precursor_per_rxn,*),
&          num_byprod_list(p_max_num_rxn,*)

real*8    rxn_data_slice(p_max_precursor_per_rxn,*)

real*8 byprod_yields(p_max_num_rxn,
&          p_max_precursor_per_rxn,*)

C
C *** locals
C
integer*4 precursors(p_max_precursor_per_rxn,p_num_site),
&          num_precursor

C --> LIST OF ADDITIONS AND SUBTRACTIONS (FOR A GIVEN RXN) WHICH
C PROVIDE A MAPPING BETWEEN THE PARENT AND PRECURSORS
integer*4 rxn_add(p_num_site),rxn_remove(p_num_site)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

C ** LOAD UP THE CURRENT REACTION UNDER CONSIDERATION
do 1002 j=1,p_num_site
  rxn_add(j) = grxn_add(rxn_idx,j)
  rxn_remove(j) = grxn_remove(rxn_idx,j)
1002 continue

C ** PERFORM ANALYSIS OF REACTIONS FOR A GIVEN PRODUCT
C I. IDENTIFY PRECURSORS
num_precursor = 0
call local_ra_phase1(product,rxn_add,rxn_remove,precursors,
&          num_precursor)

call local_ra_phase2(precursors,num_precursor,product,
& rxn_idx,rxn_group_start,rxn_group_stop,
& grxn_add,grxn_remove,byprod_list,
& byprod_card,rxn_byprod,num_byprod_list,
& byprod_yields,rxn_data_slice)

C ** UPDATE THE GLOBAL NODE LIST AND BY-PRODUCT LISTS AND ASSIGN
C POINTER LOCATIONS
call update_node_list(precursors,num_precursor,
&          gnode_indices,gnode_list,gnode_card)
C

return
end

C-----
C
C subroutine which updates the global node list and returns indices and
C the new cardinality of the list
C
subroutine update_node_list(precursors,num_precursor,
&          gnode_indices,gnode_list,gnode_card)

```

```

#include "params.f"
C
C *** args
C
    integer*4 precursors(p_max_precursor_per_rxn,*),
    &    gnode_list(p_max_molecules,*),
    &    gnode_indices(*)
    integer*4 num_precursor, gnode_card
C
C *** locals
C
C --> PRECURSOR AND GNODE MOLECULES
    integer*4 gmol(p_num_site), pmol(p_num_site)
C --> INDICATES A PRECURSOR ALREADY EXISTS IN "gnode_list"
    integer*4 hit
C --> MISCELLANEOUS INDICES
    integer*4 i,j,k
C --> UPDATED CARDINALITY
    integer*4 new_card
C*****
C
C
    new_card = gnode_card
    do 1000 i=1,num_precursor
        hit=0
C --> EXTRACT MOLECULES AND COMPARE
        do 1001 j=1,gnode_card
            do 2000 k=1,p_num_site
                pmol(k) = precursors(i,k)
                gmol(k) = gnode_list(j,k)
2000            continue
C
                if ( hit .eq. 0) then
                    call compare_mol(pmol,gmol,hit)
C --> IF "hit" = 1, STORE THE MOLECULE INDEX
                    if( hit .eq. 1) then
                        gnode_indices(i) = j
                    endif
                endif
1001            continue
                if (hit .eq. 0) then
                    new_card = new_card+1
                    gnode_indices(i) = new_card
                    do 3000 k=1,p_num_site
                        gnode_list(new_card,k) = pmol(k)
3000                continue
                endif
1000            continue
C
                call update_gnode(new_card,gnode_card)

                return
            end
C -----

```

```

C CODE FOR UPDATING 'gnode_card' (I.E., INCREMENT BY 1). CHECKS
C TO SEE IF MAXIMUM VALUE HAS BEEN SUCCEEDED; AND IF SO, TERMINATES
C WITH A STATUS MESSAGE
      subroutine update_gnode(new_card,gnode_card)
#include "params.f"
      integer*4 new_card,gnode_card
C
      if(new_card .gt. p_max_molecules)
& then
          write(*,*) '***MAX MOLECULES EXCEEDED, PROGRAM TERMINATED'
          stop
      else
          gnode_card = new_card
      endif
C
      return
      end

```

A6.2.5. Local_Ra_Phase1.f

```

C-----
C code to identify all possible precursors for a given product and reaction. The
C main inputs are product(6), rxn_add(6), and rxn_remove(6). The outputs are
C precursors(max_precursors) and num_precursor.

```

```

      subroutine local_ra_phase1(product,rxn_add,rxn_remove,precursors,
&                               num_precursor)
#include "params.f"
C
C *** args
C
      integer*4 product(*), rxn_add(*), rxn_remove(*),
&               precursors(p_max_precursor_per_rxn,*),
&               num_precursor

C
C *** locals
C
C variables used in structure representation
      integer*4 template(p_num_site),num_groups,
&             & substructure(p_max_precursor_per_rxn,p_num_site),num_hit
C temporary structure storage
      integer*4 scratch(p_num_site), mol1(p_num_site), mol2(p_num_site)
C
      integer*4 match, num_match
C miscellaneous indices
      integer*4 i,j,k,m

C*****
C initialize outputs
      num_match = 0
      num_precursor = 0

      do 1000 i=1,p_max_precursor_per_rxn
          do 1001 j=1,p_num_site
              precursors(i,j) = 0
              substructure(i,j) = 0
1001      continue
1000      continue

```

```

    do 9000 i=1,p_num_site
      template(i) = 0
      scratch(i) = 0
      mol1(i) = 0
      mol2(i) = 0
9000  continue

C
C ** construct template for identifying precursor substructures
    call get_template(rxn_add,template,num_groups)
C ** find all precursors by looking at common product structures
    call compare_mol2(rxn_add,product,template,num_groups,
      & substructure,num_hit)
C ** construct the precursors and save
C i.e., start with old structure and modify it with "rxn_remove"
    do 1002 i=1,num_hit
      do 1003 j=1,p_num_site
        precursors(i,j) = product(j)
1003  continue
1002  continue

    do 1004 i=1,num_hit
      do 1005 j=1,p_num_site
        if ( substructure(i,j) .gt. 0) then
          precursors(i,j) = rxn_remove(substructure(i,j))
        endif
1005  continue
1004  continue
C
C ** compare all structures and remove redundant ones.
    if ( num_hit .gt. 1) then
      do 2000 i=1,num_hit-num_match-1
        j = i+1
2001  if ( j .le. num_hit-num_match ) then
          do 3000 k=1,p_num_site
            mol1(k) = precursors(i,k)
            mol2(k) = precursors(j,k)
3000  continue
          match = 0
          call compare_mol(mol1,mol2,match)
          if (match .eq. 1) then
C **
            shift rows back
            do 2002 m=j+1,num_hit-num_match
              do 3002 k=1,p_num_site
                precursors(m-1,k) = precursors(m,k)
3002  continue
2002  continue
              num_match = num_match + 1
              j=j-1
            endif
            j = j + 1
            goto 2001
          endif
2000  continue

      endif

C
    call update_num_precursor(num_precursor,num_hit,num_match)
C
    return
```


end

```
-----
C subroutine to determine the location and number of groups in rxn_add vector.
C output is "template" and "num_groups"
```

```
      subroutine get_template(rxn_add,template,num_groups)
#include "params.f"
C
C *** args
C
      integer*4 rxn_add(*), template(*), num_groups

C
C *** locals
C
      integer*4 i, scratch

      num_groups = 0
      do 1000 i=1,p_num_site
        scratch = rxn_add(i)
        if(scratch .gt. 0) then
          num_groups=num_groups+1
          template(num_groups) = i
        endif
1000 continue
      return
      end
```

```
-----
C CODE FOR UPDATING THE NUMBER OF PRECURSORS PER REACTION. CHECKS
C TO SEE IF MAXIMUM LEVEL HAS BEEN SUCCEEDED; AND IF SO, TERMINATES
C WITH A STATUS MESSAGE. THE LOCATION OF THIS CODE IS SUCH THAT IT
C IS NOT SUFFICIENT FOR STOPPING EVERY ARRAY VIOLATION BEFORE TERMINATION,
C BUT IF A VIOLATION OCCURS, THIS CODE WILL IDENTIFY IT RETROACTIVELY AND
C THEN TERMINATE. THIS IS DONE FOR COMPUTATIONAL EFFICACY.
```

```
      subroutine update_num_precursor(num_precursor,num_hit,num_match)
#include "params.f"
      integer*4 num_precursor, num_hit, num_match

C
      num_precursor = num_hit-num_match
      if(num_precursor .gt. p_max_precursor_per_rxn) then
        write(*,*) '***MAX PRECURSORS EXCEEDED FOR A RXN'
        stop
      endif

C
      return
      end
```

A6.2.6. Local_Ra_Phase2.f

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C LOCAL_RA_PHASE2 (FORTRAN 77)
C Given the rxn, precursor and product, this code
C evaluates the by-products and performs quantitative
C analysis.
C
```

```

C Coded by and property of Jonathan P. Knight
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
  subroutine local_ra_phase2(precursors,num_precursor,
    & product,rxn_idx,rxn_group_start,rxn_group_stop,
    & grxn_add,grxn_remove,byprod_list,
    & byprod_card,rxn_byprod,num_byprod_list,
    & byprod_yields,rxn_data_slice)
#include "params.f"
C
C *** ARGS
C
  integer*4
  & precursors(p_max_precursor_per_rxn,*),num_precursor,
  & rxn_idx,rxn_group_start,rxn_group_stop,
  & byprod_list(p_max_molecules,*),byprod_card,
  & product(*),grxn_add(p_max_num_rxn,*),
  & grxn_remove(p_max_num_rxn,*)

  integer*4 rxn_byprod(p_max_num_rxn,
  & p_max_precursor_per_rxn,*),
  & num_byprod_list(p_max_num_rxn,*)

  real*8 rxn_data_slice(p_max_precursor_per_rxn,*)

  real*8 byprod_yields(p_max_num_rxn,
  & p_max_precursor_per_rxn,*)
C
C *** LOCALS
C
C --> PRECURSOR TO BE EVALUATED
  integer*4 rm(p_num_site),
  & wt(p_max_precursor_per_rxn),
  & byprods(p_max_precursor_per_rxn,p_num_site),
  & num_byproduct

C --> THE LOCATION OF SITES WHICH MUST BE INTERCHANGED AS A
C RESULT OF A CHEMICAL REACTION. THE STRUCTURE IS ONE IN
C WHICH NONZERO LOCATIONS INDICATE A REACTION SITE. THESE
C NONZERO LOCATIONS POINT INTO THE "rxn_add" ARRAY.
  integer*4 substructure(p_max_precursor_per_rxn,p_num_site)

C --> INDICES OF BYPRODUCTS INTO GLOBAL LIST
  integer*4 byprod_indices(p_max_precursor_per_rxn)

C --> A STRIP FROM "substructures" AND "wt" FOR A PARTICULAR
C BYPRODUCT
  integer*4 site(p_num_site), num_equiv

C --> RATE FOR A PRODUCT OR BYPRODUCT OF A CHEMICAL REACTION,
C THE ACCUMULATED LIST OF SUCH RATES (FOR A SINGLE REACTION)
C AND THE RATE OF THE SINGLE DESIRED PRODUCTS
  real*8 any_rate, rxn_rates(p_max_precursor_per_rxn),
  & product_rate(p_max_precursor_per_rxn)

C --> BYPRODUCT YIELDS FOR REACTIONS, AND MAIN YIELD
  real*8 yields(p_max_precursor_per_rxn), prod_yield

C --> LOCATION OF PRODUCT IN "byprods" ARRAY. PRODUCT ID IS USED
C TO GET THE RATE OF FORMATION OF THE PRODUCT.
  integer*4 product_id

```

```

C --> CURRENT REACTION UNDER CONSIDERATION FOR THE GENERATION OF
C BY-PRODUCTS
      integer*4 rxn_add(p_num_site), rxn_remove(p_num_site)

C --> NEWLY GENERATED BYPRODUCTS
      integer*4 new_byprods(p_max_precursor_per_rxn,
&      p_num_site),
&      new_num_byproduct

C --> NEWLY GENERATED SUBSTRUCTURES
      integer*4 new_substructure(p_max_precursor_per_rxn,
&      p_num_site)

C --> LIST OF REACTION POINTERS CORRESPONDING TO EACH OF THE
C BYPRODUCTS.
      integer*4 saved_idx(p_max_precursor_per_rxn)

C --> OVERALL RATE OF FORMATION OF THE DESIRED PRODUCT
      real*8 scratch_rate

C --> MISC. INDICES
      integer*4 i,j,k,m,idx,scratch_idx

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

C ** FOR EACH PRECURSOR PERFORM THE REACTION ANALYSIS
C TO ASSESS RATE,YIELD AND PRODUCT DISTRIBUTIONS
      do 1000 i=1,num_precursor
C ** EXTRACT THE PRECURSOR STRUCTURE INTO A "RAW MATERIAL"
C ARRAY
      do 2000 j=1,p_num_site
          rm(j) = precursors(i,j)
2000      continue

C ** OBTAIN ALL BYPRODUCTS FOR A REACTION AND ADD THEM
C TO THE LIST OF BYPRODUCTS
C 0. INITIALIZE THE NUMBER OF BYPRODUCTS FOR A PRECURSOR
C TO A NULL VALUE AND THE PRODUCT FORMATION RATE
C I. SELECT A REACTION FROM THE GROUPING
C II. DETERMINE THE BY-PRODUCTS GENERATED BY THAT
C REACTION
C III. UPDATE THE BYPRODUCT LIST AND NEW SUBSTRUCTURES
C IV. SAVE THE REACTIONS IDS CORRESPONDING TO EACH BYPRODUCT
C V. UPDATE THE NUMBER OF BYPRODUCT

C ** (0)
      num_byproduct = 0
      rxn_data_slice(i,p_rxn_rate) = 0.0

C *****
C ** BEGIN INNER LOOP
C *****

      do 5001 idx = rxn_group_start,rxn_group_stop

C ** (I)
      do 4112 j=1,p_num_site
          rxn_add(j) = grxn_add (idx,j)
          rxn_remove(j) = grxn_remove(idx,j)
4112      continue

```

```

C **      (II)
      call get_byprods (rm,rxn_remove,rxn_add,
&          new_byprods,new_num_byproduct,
&          new_substructure)

C **      (III)
      do 4120 j=1,new_num_byproduct
        do 4121 k=1,p_num_site
          byprods(num_byproduct+j,k)=
&          new_byprods(j,k)
          substructure(num_byproduct+j,k)=
&          new_substructure(j,k)
4121      continue
4120      continue

C **      (IV)
      do 5111 k=1, new_num_byproduct
        scratch_idx = k + num_byproduct
        saved_idx(scratch_idx)=idx
5111      continue

C **      (V)
      num_byproduct = num_byproduct + new_num_byproduct

C *****
C **      END INNER LOOP
C *****
5001      continue

C ** ENTER A NEW INNER LOOP TO DETERMINE THE RATES OF REACTION
C      FOR EACH BYPRODUCT
C *****
C **      ENTER INNER LOOP
C *****

      num_equiv = 1
      do 2001 k=1, num_byproduct
C **      DETERMINE THE CURRENT REACTION SITE
        do 3000 m = 1,p_num_site
          site(m) = substructure(k,m)
3000      continue

C **      DETERMINE THE CURRENT REACTION INDEX
        idx      = saved_idx (k)

C **      COMPUTE AND STORE THE RATE OF THE REACTION
        call get_rate_data(site,num_equiv,rm,idx,
&          any_rate)
        rxn_rates(k) = any_rate

C *****
C **      EXIT INNER LOOP
C *****
2001      continue

C **      FIND THE UNIQUE SET OF BY-PRODUCTS AND THE AGGREGATE RATES
C      YOU NEED TO UPDATE THE SAVED IDX'S
&      call compress_byprods(byprods,num_byproduct,
&          substructure,product,scratch_rate,product_id,

```

```

&          rxn_rates)

C **      SAVE THE AGGREGATE RATE CORRESPONDING TO THE PRODUCT
          rxn_data_slice(i,p_rxn_rate) = scratch_rate

C **      GET THE YIELDS
          call get_rxn_data(rxn_rates,num_byproduct,product_id,
&          yields,prod_yield)

C **      SAVE THE YIELDS
          rxn_data_slice(i,p_rxn_yield) = prod_yield
          do 4001 k=1,num_byproduct
            byprod_yields(rxn_idx,i,k) = yields(k)
4001      continue

C **      UPDATE THE CURRENT GLOBAL LIST OF BYPRODUCTS
          call update_byprod_list(byprods,num_byproduct,
&          byprod_indices,byprod_list,byprod_card)

C **      UPDATE THE CURRENT BYPRODUCT STACK
          num_byprod_list(rxn_idx,i) = num_byproduct
          do 4000 k=1,num_byproduct
            rxn_byprod(rxn_idx,i,k) = byprod_indices(k)
4000      continue

1000  continue
C
      return
      end

C -----
C COMPRESS_BYPRODS (FORTRAN 77)
C A code to form a unique list of byproducts and rates
C
C Coded by and property of Jonathan P. Knight
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      subroutine compress_byprods(byprods,num_byprod,
&          substructure,product,product_rate,
&          product_id,rxn_rates)

#include "params.f"

C
C *** ARGS
C
      integer*4 byprods(p_max_precursor_per_rxn,*),
&          num_byprod,
&          substructure(p_max_precursor_per_rxn,*),
&          product(*),product_id
      real*8   rxn_rates(*),product_rate

C
C *** LOCALS
C
C --> NUMBER OF BY-PRODUCTS GENERATED BY THE REACTION (NOT NECESSARILY UNIQUE)
      integer*4 num_hit

```

```

C --> TEMPORARY STORAGE FOR MOLECULES
      integer*4 mol1(p_num_site), mol2(p_num_site)
C
      integer*4 match, num_match, hit

C --> MISC INDICES
      integer*4 i, j, k, m

      num_hit = num_byprod
      num_match = 0

C
C ** COMPARE ALL STRUCTURES AND REMOVE REDUNDANT ONES
      if ( num_hit .gt. 1) then
        do 2000 i=1, num_hit-num_match-1
          j = i+1
2001      if ( j .le. num_hit-num_match ) then
            do 3000 k=1, p_num_site
              mol1(k) = byprods(i, k)
              mol2(k) = byprods(j, k)
3000      continue
            match = 0
            call compare_mol(mol1, mol2, match)
            if (match .eq. 1) then
              rxn_rates(i) = rxn_rates(i)+rxn_rates(j)
C **      shift rows back
              do 2002 m=j+1, num_hit-num_match
                do 3002 k=1, p_num_site
                  rxn_rates(m-1)=rxn_rates(m)
                  byprods(m-1, k) = byprods(m, k)
                  substructure(m-1, k) =
                    &      substructure(m, k)
3002      continue
2002      continue
            num_match = num_match + 1
            j=j-1
            endif
            j = j + 1
            goto 2001
          endif
2000      continue

        endif

C ** † ACCOUNT FOR THE COMPRESSED NUMBER OF BYPRODUCTS
      call update_num_byprod(num_byprod, num_hit, num_match)

C ** IDENTIFY THE LOCATION OF PRODUCT IN LIST
      do 5100 i =1, num_byprod
        hit=0
        do 5101 j = 1, p_num_site
          mol1(j) = byprods(i, j)
5101      continue
          call compare_mol(mol1, product, hit)
          if(hit .eq. 1) then
            product_rate = rxn_rates(i)
            product_id = i
          endif
5100      continue
C
      return

```

end

A6.2.7. Evaluate_Rxn_Path.f

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C EVALUATE_RXN_PATH (FORTRAN 77)
C
C UTILIZES DATA STACKS TO DETERMINE IF INCREMENTAL AST IS
C FEASIBLE IN THE CONTEXT OF THE CURRENT PATHWAY. IT IS
C IMPORTANT TO NOTE THAT THIS CODE WILL NOT CATCH THE ROOT
C NODE AS A "PATH", SO IF MIN PATH YIELD = 1.2, ONLY THE
C ROOT NODES WILL BE DISPLAYED, DEPENDING ON THE SINGLE STEP
C YIELDS
C
C coded by and property of jonathan knight
C dept. chemical engineering
C MIT
C
C created 02/06,93
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      subroutine evaluate_rxn_path(parent_level,num_rxn,
&      curr_node,rxn_ptr_stack,rxn_ptr_list,
&      rxn_data_stack,rxn_node_ptr,
&      num_precursor_list,rxn_data,
&      child_node_stack,num_child_nodes)
#include "params.f"

C
C *** ARGS
C
      integer*4 parent_level,num_rxn,curr_node(*)

      integer*4 rxn_ptr_stack(p_max_level,
& p_max_num_rxn,*),
& rxn_ptr_list(p_max_level,*)

      real*8 rxn_data_stack(p_max_level,
& p_max_num_rxn,p_max_precursor_per_rxn,*)

      integer*4 rxn_node_ptr(p_max_num_rxn,*),
& num_precursor_list(*)

      real*8 rxn_data(p_max_num_rxn,p_max_precursor_per_rxn,
& *)

      integer*4 child_node_stack(*),
& num_child_nodes

C
C *** LOCALS
C
C --> MAXIMUM YIELD FOR ALL REACTIONS POINTING TO A GIVEN
C NODE
      real*8 max_yield

C --> YIELD WHOSE VALUE IS TO BE COMPARED WITH TEST YIELD
      real*8 test_yield

C --> NODE WHOSE YIELD VALUE IS TO BE COMPARED
      integer*4 test_node

```

```

C --> NET YIELD FOR A PATHWAY
      real*8 net_yield

C --> YIELD FOR NEW STEP
      real*8 new_yield

C --> MISC. INDICES
      integer*4 i,j,k

C ** FOR EACH LEVEL, FIND ALL REACTIONS WHICH POINT TO THE
C    CURRENT NODE IN THE STACK, AND COMPUTE THE YIELD MEASURE
      net_yield=1.0
      do 1000 i=1,parent_level

C ** DEBUGGING INFO
C    write(*,*) 'level = ',i
      max_yield=0.0
      do 1001 j=1,num_rxn
        do 1002 k=1,rxn_ptr_list(i,j)
          test_node=rxn_ptr_stack(i,j,k)
          if(test_node .eq. curr_node(i)) then
            test_yield=rxn_data_stack(i,j,k,
            &          p_rxn_yield)
            max_yield=max(max_yield,test_yield)
          endif
1002      continue
1001      continue
          net_yield=net_yield*max_yield

C ** DEBUGGING INFO
C    write(*,*) '-->',net_yield
1000      continue

C ** IDENTIFY PATHWAYS WHICH HAVE BEEN MADE INFEASIBLE BY
C    THE NEW REACTION AND ELIMINATE THEM BY SETTING THE
C    RXN POINTER TO A NULL VALUE (IT CAN'T POINT TO ANY
C    NODE
      do 2000 i=1,num_rxn
        new_yield=0.0
        do 2001 j=1,num_precursor_list(i)
          new_yield=rxn_data(i,j,p_rxn_yield)
          new_yield=net_yield*new_yield
          if(new_yield .lt. p_min_path_yield)
            &          rxn_node_ptr(i,j) = p_null_value
2001      continue
2000      continue

C ** FINALLY, ELIMINATE ANY PRECURSORS WHO DO NOT HAVE AT
C    LEAST ONE FAVORABLE PATH
      call phase2_pruning(num_child_nodes,child_node_stack,
      &          num_rxn,num_precursor_list,rxn_node_ptr)
C
      return
      end

```

A6.2.8. Get_Byprods.f

```

C-----
C code to identify all byproducts for a given product and reaction. The
C main inputs are product(6), rxn_remove(6), and rxn_add(6). The outputs are
C byprods(max_byprods) and num_byprod.

```



```

C
C THIS CODE DOES NOT RECOGNIZE ALL POSSIBLE SYMMETRIES.  FOR EXAMPLE, TWO
C BYPRODUCTS ARE GENERATED FOR THE REACTION <5 0 5 0 0 0> --> 2 <3 0 3 0 0 0>

      subroutine get_byprods (rm,rxn_remove,rxn_add,
&
&      byprods,
&
&      num_byprod,substructure)
#include "params.f"
C
C *** args
      integer*4 rm(*), rxn_remove(*), rxn_add(*),
&
&      byprods(p_max_precursor_per_rxn,*),
&
&      num_byprod,
&
&      substructure(p_max_precursor_per_rxn,*)

C
C *** locals
C
C --> LOCATION OF GROUPS TO BE INTERCHANGED WITHIN "rxn_remove" AND "rxn_add"
C      ARRAYS, THE TOTAL NUMBER OF GROUPS THAT ARE REMOVED
      integer*4 template(p_num_site),num_groups

C --> NUMBER OF BY-PRODUCTS GENERATED BY THE REACTION (NOT NECESSARILY UNIQUE)
      integer*4 num_hit

C --> TEMPORARY STORAGE FOR MOLECULES
      integer*4 scratch(p_num_site)

C --> MISC INDICES
      integer*4 i,j,k,m

C*****
C initialize outputs
      num_match = 0
      num_byprod = 0

      do 1000 i=1,p_max_precursor_per_rxn
        do 1001 j=1,p_num_site
          byprods(i,j) = 0
          substructure(i,j) = 0
1001      continue
1000      continue

      do 9000 i=1,p_num_site
        template(i) = 0
        scratch(i) = 0
9000      continue

C
C ** CONSTRUCT THE TEMPLATE FOR IDENTIFYING BYPRODUCT MOLECULES
      call get_template(rxn_remove,template,num_groups)

C ** FIND ALL BYPRODUCTS BY LOOKING AT COMMON SUBSTRUCTURES
      call compare_mol2(rxn_remove,rm,template,num_groups,
&
&      substructure,num_hit)

C ** CONSTRUCT THE BYPRODUCTS AND SAVE
C      i.e., start with old structure and modify it with "rxn_add"
      do 1002 i=1,num_hit
        do 1003 j=1,p_num_site

```

```

        byprods(i,j) = rm(j)
1003   continue
1002  continue

      do 1004 i=1,num_hit
        do 1005 j=1,p_num_site
          if ( substructure(i,j) .gt. 0) then
            byprods(i,j) = rxn_add(substructure(i,j))
          endif
1005   continue
1004  continue
C
      num_byprod = num_hit
C
      return
      end

C -----
C CODE FOR UPDATING THE NUMBER OF PRECURSORS PER REACTION. CHECKS
C TO SEE IF MAXIMUM LEVEL HAS BEEN SUCCEEDED; AND IF SO, TERMINATES
C WITH A STATUS MESSAGE. THE LOCATION OF THIS CODE IS SUCH THAT IT
C IS NOT SUFFICIENT FOR STOPPING EVERY ARRAY VIOLATION BEFORE TERMINATION,
C BUT IF A VIOLATION OCCURS, THIS CODE WILL IDENTIFY IT RETROACTIVELY AND
C THEN TERMINATE. THIS IS DONE FOR COMPUTATIONAL EFFICACY.
      subroutine update_num_byprod(num_byprod,num_hit,num_match)
#include "params.f"
      integer*4 num_byprod, num_hit, num_match
C
      num_byprod = num_hit-num_match
      if(num_byprod .gt. p_max_precursor_per_rxn) then
        write(*,*) '***MAX BYPRODUCTS EXCEEDED FOR A RXN'
        stop
      endif
C
      return
      end

```

A6.2.9. Get_Rate_Data.f

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C GET_RATE_DATA (FORTRAN 77)
C
C coded by and property of jonathan P. knight
C dept. chemical engineering
C MIT
C
C created 02/02/1993
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      subroutine get_rate_data(site,num_equiv,rm,
        & rxn_idx,rrate)
#include "params.f"
C
C *** ARGS
C
      integer*4 site(*),num_equiv,rm(*),rxn_idx
      real*8 rrate
C

```

```
C *** LOCALS
C
C --> LOCATION OF REACTION CENTER
      integer*4 site_idx
C --> O/M/P LOCATIONS
      integer*4 o1,o2,m1,m2,p
C --> O/M/P RATE CONTRIBUTIONS
      real*8 ro1,ro2,rml,rm2,rp
C --> RATE COEFFICIENTS
      real*8 prateso(p_max_num_rxn,p_max_substit),
      &      pratesm(p_max_num_rxn,p_max_substit),
      &      pratesp(p_max_num_rxn,p_max_substit)
C --> SANDEMEYER RATE
      real*8 sandmeyer_rate
C --> BECHAMP RATE
      real*8 bechamp_rate
C --> RATE FOR DE-AMINO-HYDROGENATION
      real*8 deamino_rate
C --> NUMBER OF NITROS AND AMINOS ON RAW MATERIAL
      integer*4 nitros, aminos, chloro, total
C --> MISC. INDICES
      integer*4 i,count
C ** LOAD UP THE SUBSTITUENT COEFFICIENTS
C ** SUBSTITUENTS:
C   1 = H
C   2 = Cl
C   3 = CN
C   4 = NO2
C   5 = NH2
C ** FIRST 16 REACTIONS: SEE BELOW
C ** REACTION 17: NITRATION
      prateso(17,1) = 1.0
      prateso(17,2) = 0.03
      prateso(17,3) = 9.28E-5
      prateso(17,4) = 2.88E-6
      prateso(17,5) = 8921.0
      prateso(17,6) = 0.0
      prateso(17,7) = 0.0
      prateso(17,8) = 0.0
      pratesm(17,1) = 1.0
      pratesm(17,2) = 8.4E-4
      pratesm(17,3) = 4.36E-4
      pratesm(17,4) = 4.17E-5
      pratesm(17,5) = 0.33
      pratesm(17,6) = 0.0
      pratesm(17,7) = 0.0
```

pratesm(17,8) = 0.0

pratesp(17,1) = 1.0

pratesp(17,2) = 0.13

pratesp(17,3) = 2.15E-5

pratesp(17,4) = 2.69E-7

pratesp(17,5) = 5248.0

pratesp(17,6) = 0.0

pratesp(17,7) = 0.0

pratesp(17,8) = 0.0

C ** REACTION 18: NAS OF CHLORINE BY AMMONIA.

C (ACTIVATED BY NITRO- OR CYANO- GROUP)

prateso(18,1) = 1.0

prateso(18,2) = 1.57

prateso(18,3) = 18.4

prateso(18,4) = 251.0

prateso(18,5) = 1.0E-5

prateso(18,6) = 0.0

prateso(18,7) = 0.0

prateso(18,8) = 0.0

pratesm(18,1) = 1.0

pratesm(18,2) = 3.03

pratesm(18,3) = 3.6

pratesm(18,4) = 5.0

pratesm(18,5) = 1.0E-5

pratesm(18,6) = 0.0

pratesm(18,7) = 0.0

pratesm(18,8) = 0.0

pratesp(18,1) = 1.0

pratesp(18,2) = 1.57

pratesp(18,3) = 20.1

pratesp(18,4) = 45.2

pratesp(18,5) = 1.0E-5

pratesp(18,6) = 0.0

pratesp(18,7) = 0.0

pratesp(18,8) = 0.0

C ** REACTION 19: SUBSTITUTIVE CHLORINATION

C

prateso(19,1) = 1.0

prateso(19,2) = 0.019

prateso(19,3) = 5.38E-7

prateso(19,4) = 6.28E-9

prateso(19,5) = 6.1E5

prateso(19,6) = 0.0

prateso(19,7) = 0.0

prateso(19,8) = 0.0

pratesm(19,1) = 1.0

pratesm(19,2) = 6.54E-4

pratesm(19,3) = 1.71E-6

pratesm(19,4) = 3.03E-8

pratesm(19,5) = 0.15

pratesm(19,6) = 0.0

pratesm(19,7) = 0.0

pratesm(19,8) = 0.0

```

pratesp(19,1) = 1.0
pratesp(19,2) = 0.0654
pratesp(19,3) = 1.35E-7
pratesp(19,4) = 1.12E-9
pratesp(19,5) = 2.5E6
pratesp(19,6) = 0.0
pratesp(19,7) = 0.0
pratesp(19,8) = 0.0

C ** FIND REACTION SITE
  do 1000 i=1,p_num_site
    if(site(i) .eq. 1) site_idx=i
1000  continue

C ** FIND O/M/P SITES AND THEIR IDENTITY
o1 = site_idx+1
if(o1 .gt. 6) o1=o1-6
ro1=prateso(rxn_idx,rm(o1))

o2 = site_idx-1
if(o2 .lt. 1) o2=o2+6
ro2=prateso(rxn_idx,rm(o2))

m1 = site_idx+2
if(m1 .gt. 6) m1=m1-6
rm1=pratesm(rxn_idx,rm(m1))

m2 = site_idx-2
if(m2 .lt. 1) m2=m2+6
rm2=pratesm(rxn_idx,rm(m2))

p = site_idx+3
if(p .gt. 6) p=p-6
rp=pratesp(rxn_idx,rm(p))

rrate = ro1*ro2*rm1*rm2*rp*dble(num_equiv)

C ** REACTION 1: DIAZOTIZATION + SANDMEYER REACTION (NH2 -> CN)
C   I. FIRST, CHECK FOR DIAMINES. IF THERE ARE ANY, SET
C     REACTION RATE TO ZERO.
C
C   II. SUBSTITUENT EFFECTS ARE NEGLIGIBLE IN THE OVERALL REACTION
C     FOR THE SUBSTITUENTS WE CONSIDER. SET 'rrate' DIRECTLY.

  if(rxn_idx .eq. 1) then
    sandmeyer_rate = 1.0
    count = 0
    do 3400 i=1,p_num_site
      if(rm(i) .eq. 5) count=count+1
3400  continue
    if(count .gt. 1) sandmeyer_rate = 0.0
    rrate = sandmeyer_rate
  endif

C ** REACTION 2: DIAZONOTATION + SANDMEYER REACTION BYPRODUCT (NH2 -> c-NNCN)
C   I. FIRST, CHECK FOR DIAMINES. IF THERE ARE ANY, SET REACTION RATE
C     TO ZERO.
C
C
  if(rxn_idx .eq. 2) then

```

```

    sandmeyer_ratio = 0.2
    sandmeyer_rate = 1.0
    count = 0
    do 3401 i=1,p_num_site
        if(rm(i).eq. 5) count = count+1
3401    continue
        if(count .gt. 1) sandmeyer_rate = 0.0
        rrate = sandmeyer_rate*sandmeyer_ratio/
&    (1.0+sandmeyer_ratio)
    endif

C ** REACTION 3: SAME AS REACTION 1 EXCEPT TWO META AMINES ARE REACTING.
C    m-(NH2,NH2) -> m-(CN,CN)
C    FIRST, MAKE SURE THERE THE SYSTEM IS NOT TRI-SUBSTITUTED. THE
C    REACTION RATE IS SCALED TO ACCOUNT FOR THE SYMMETRY WHICH THE BY-PRODUCT
C    GENERATOR CURRENTLY FAILS TO DETECT.
    if(rxn_idx .eq. 3) then
        sandmeyer_rate = 1.0
        sandmeyer_ratio = 0.2
        count = 0
        do 3402 i=1,p_num_site
            if(rm(i).eq. 5) count = count+1
3402    continue
            if(count .gt. 2) sandmeyer_rate = 0.0
            rrate = sandmeyer_rate/(sandmeyer_ratio+1.0)**2
        endif

C ** REACTION 4: DIAZONOTATION + SANDMEYER REACTION BYPRODUCT
C    {m-(NH2,NH2) -> m-(c-NNCN,c-NNCN)}
C    I. FIRST, CHECK FOR TRIAMINES. IF THERE ARE ANY, SET REACTION RATE
C        TO ZERO.
C    REACTION RATE IS SCALED TO ACCOUNT FOR THE SYMMETRY WHICH THE BY-PRODUCT
C    GENERATOR CURRENTLY FAILS TO DETECT.
C
    if(rxn_idx .eq. 4) then
        sandmeyer_ratio = 0.2
        sandmeyer_rate = 1.0
        count = 0
        do 3403 i=1,p_num_site
            if(rm(i).eq. 5) count = count+1
3403    continue
            if(count .gt. 2) sandmeyer_rate = 0.0
            rrate = sandmeyer_rate*sandmeyer_ratio**2
&            /(1.0+sandmeyer_ratio)**2
        endif

C ** REACTION 5: DIAZONOTATION + SANDMEYER REACTION BYPRODUCT
C    {m-(NH2,NH2) -> (CN,c-NNCN)}
C    I. FIRST, CHECK FOR TRIAMINES. IF THERE ARE ANY, SET REACTION RATE
C        TO ZERO.
C
    if(rxn_idx .eq. 5) then
        sandmeyer_ratio = 0.2
        sandmeyer_rate = 1.0
        count = 0
        do 3404 i=1,p_num_site
            if(rm(i).eq. 5) count = count+1
3404    continue
            if(count .gt. 2) sandmeyer_rate = 0.0
            rrate = sandmeyer_rate*sandmeyer_ratio
&            /(1.0+sandmeyer_ratio)**2
```

```
endif

C ** REACTION 6: SAME AS REACTION 1 EXCEPT TWO PARA AMINES ARE REACTING.
C   p-(NH2,NH2) -> p-(CN,CN)
C   FIRST, MAKE SURE THERE THE SYSTEM IS NOT TRI-SUBSTITUTED
C   if(rxn_idx .eq. 6) then
C     sandmeyer_rate = 1.0
C     sandmeyer_ratio = 0.2
C     count = 0
C     do 3405 i=1,p_num_site
C       if(rm(i) .eq. 5) count=count+1
3405    continue
C     if(count .gt. 2) sandmeyer_rate = 0.0
C     rrate = sandmeyer_rate/(sandmeyer_ratio+1.0)**2
C   endif

C ** REACTION 7: DIAZONOTATION + SANDMEYER REACTION BYPRODUCT
C   {p-(NH2,NH2) -> p-(CN,c-NNCN)}
C   I. FIRST, CHECK FOR TRIAMINES. IF THERE ARE ANY, SET REACTION RATE
C     TO ZERO.
C
C   if(rxn_idx .eq. 7) then
C     sandmeyer_ratio = 0.2
C     sandmeyer_rate = 1.0
C     count = 0
C     do 3406 i=1,p_num_site
C       if(rm(i) .eq. 5) count=count+1
3406    continue
C     if(count .gt. 2) sandmeyer_rate = 0.0
C     rrate = sandmeyer_rate*sandmeyer_ratio**2
C     &      /((1.0+sandmeyer_ratio)**2)
C   endif

C ** REACTION 8: DIAZONOTATION + SANDMEYER REACTION BYPRODUCT
C   {p-(NH2,NH2) -> p-(CN,c-NNCN)}
C   I. FIRST, CHECK FOR TRIAMINES. IF THERE ARE ANY, SET REACTION RATE
C     TO ZERO.
C
C   if(rxn_idx .eq. 8) then
C     sandmeyer_ratio = 0.2
C     sandmeyer_rate = 1.0
C     count = 0
C     do 3407 i=1,p_num_site
C       if(rm(i) .eq. 5) count=count+1
3407    continue
C     if(count .gt. 2) sandmeyer_rate = 0.0
C     rrate = sandmeyer_ratio
C     &      /((1.0+sandmeyer_ratio)**2)
C   endif

C ** REACTION 9: SAME AS REACTION 1 EXCEPT A TRISUBSTITUTED BENZENE IS REACTING.
C   FIRST, MAKE SURE THERE THE SYSTEM IS NOT TETRA-SUBSTITUTED
C   if(rxn_idx .eq. 9) then
C     sandmeyer_rate = 1.0
C     sandmeyer_ratio = 0.2
C     count = 0
C     do 3408 i=1,p_num_site
C       if(rm(i) .eq. 5) count=count+1
```

```
3408     continue
        if(count .gt. 3) sandmeyer_rate = 0.0
        rrate = sandmeyer_rate/(1.0+sandmeyer_ratio)**3
    endif

C ** REACTION 10: DIAZONOTATION + SANDMEYER REACTION BYPRODUCT
C   I.  FIRST, CHECK FOR TETRA-AMINES.  IF THERE ARE ANY, SET REACTION RATE
C       TO ZERO.
C
    if(rxn_idx .eq. 10) then
        sandmeyer_ratio = 0.2
        sandmeyer_rate = 1.0
        count = 0
        do 3409 i=1,p_num_site
            if(rm(i) .eq. 5) count=count+1
3409     continue
            if(count .gt. 3) sandmeyer_rate = 0.0
            rrate = sandmeyer_rate*sandmeyer_ratio**3/
                & (1.0+sandmeyer_ratio)**3
        endif

C ** REACTION 11: DIAZONOTATION + SANDMEYER REACTION BYPRODUCT
C   I.  FIRST, CHECK FOR TETRA-AMINES.  IF THERE ARE ANY, SET REACTION RATE
C       TO ZERO.
C
    if(rxn_idx .eq. 11) then
        sandmeyer_ratio = 0.2
        sandmeyer_rate = 1.0
        count = 0
        do 3410 i=1,p_num_site
            if(rm(i) .eq. 5) count=count+1
3410     continue
            if(count .gt. 3) sandmeyer_rate = 0.0
            rrate = sandmeyer_rate*sandmeyer_ratio**2/
                & (1.0+sandmeyer_ratio)**3
        endif

C ** REACTION 12: DIAZONOTATION + SANDMEYER REACTION BYPRODUCT
C   I.  FIRST, CHECK FOR TETRA-AMINES.  IF THERE ARE ANY, SET REACTION RATE
C       TO ZERO.
C
    if(rxn_idx .eq. 12) then
        sandmeyer_ratio = 0.2
        sandmeyer_rate = 1.0
        count = 0
        do 3411 i=1,p_num_site
            if(rm(i) .eq. 5) count=count+1
3411     continue
            if(count .gt. 3) sandmeyer_rate = 0.0
            rrate = sandmeyer_rate*sandmeyer_ratio/
                & (1.0+sandmeyer_ratio)**3
        endif

C ** REACTION 13: BECHAMP REDUCTION
C   I.  FIRST, CHECK FOR DINITROS.  IF THERE ARE ANY, SET
C       REACTION RATE TO ZERO.
C
C   II. SUBSTITUENT EFFECTS ARE NEGLIGIBLE IN THE OVERALL REACTION
C       FOR THE SUBSTITUENTS WE CONSIDER.  SET 'rrate' DIRECTLY.
```



```
    if(rxn_idx .eq. 13) then
      bechamp_rate = 1.0
      count = 0
      do 3412 i=1,p_num_site
        if(rm(i) .eq. 4) count=count+1
3412      continue
      if(count .gt. 1) bechamp_rate = 0.0
      rrate = bechamp_rate
    endif

C ** REACTION 14: BECHAMP REDUCTION (TWO NITROS)
C   I.  FIRST, CHECK FOR TRINITROS.  IF THERE ARE ANY, SET
C       REACTION RATE TO ZERO.
C
C   II. SUBSTITUENT EFFECTS ARE NEGLIGIBLE IN THE OVERALL REACTION
C       FOR THE SUBSTITUENTS WE CONSIDER.  SET 'rrate' DIRECTLY.

    if(rxn_idx .eq. 14 .or. rxn_idx .eq. 15 .or.
& rxn_idx .eq. 16) then
      bechamp_rate = 1.0
      count = 0
      do 3413 i=1,p_num_site
        if(rm(i) .eq. 4) count=count+1
3413      continue
      if(count .gt. 2) bechamp_rate = 0.0
      rrate = bechamp_rate
    endif

C ** RATE SCALING FOR REACTION 18
C   if(rxn_idx .eq. 18) rrate=rrate/37000.0

C ** REACTION 24-27: CONVERSION OF AMINO TO HYDROGEN VIA DIAZOTIZATION
C   I.  FIRST, CHECK FOR DIAMINES, TRIAMINES ETC.  IF THERE ARE ANY, SET
C       REACTION RATE TO ZERO.
C
C   II. SUBSTITUENT EFFECTS ARE NEGLIGIBLE IN THE OVERALL REACTION
C       FOR THE SUBSTITUENTS WE CONSIDER.  SET 'rrate' DIRECTLY.

    if(rxn_idx .eq. 24) then
      deamino_rate = 1.0
      count = 0
      do 3611 i=1,p_num_site
        if(rm(i) .eq. 5) count=count+1
3611      continue
      if(count .gt. 1) deamino_rate = 0.0
      rrate = deamino_rate
    endif

    if(rxn_idx .eq. 25) then
      deamino_rate = 1.0
      count = 0
      do 3612 i=1,p_num_site
        if(rm(i) .eq. 5) count=count+1
3612      continue
      if(count .gt. 2) deamino_rate = 0.0
      rrate = deamino_rate
    endif

    if(rxn_idx .eq. 26) then
      deamino_rate = 1.0
```

```

        count = 0
        do 3613 i=1,p_num_site
            if(rm(i) .eq. 5) count=count+1
3613    continue
            if(count .gt. 2) deamino_rate = 0.0
            rrate = deamino_rate
        endif

        if(rxn_idx .eq. 27) then
            deamino_rate = 1.0
            count = 0
            do 3614 i=1,p_num_site
                if(rm(i) .eq. 5) count=count+1
3614    continue
                if(count .gt. 2) deamino_rate = 0.0
                rrate = deamino_rate
            endif

C ** CHECK THE SUBSTRATE TO SEE IF THERE ARE MULTIPLE NITRO OR
C AMINO GROUPS. IF SO, SET THE RATE TO ZERO.
        nitros=0
        aminos=0
        chloro=0
        total=0
        do 555 ii=1,p_num_site
            if(rm(ii) .eq. 4) nitros=nitros+1
            if(rm(ii) .eq. 5) aminos=aminos+1
            if(rm(ii) .eq. 2) chloro=chloro+1
            if(rm(ii) .ne. 1) total=total+1
555    continue

        if( rxn_idx .eq. 18) then
            rrate=1.
            if(
&    ( rm(o1) .ne. 4 .and. rm(o2) .ne. 4 .and.
&    rm(o1) .ne. 4 .and. rm(o2) .ne. 4) .and.
&    (rm(p) .ne. 4 .and. rm(p) .ne. 4)
&    ) rrate = 0.0
            endif

            if((nitros .gt. 1 .or. aminos .gt. 1) .or.
&    (rxn_idx .eq. 24 .and. total .lt. -1))
&    rrate = 0.0

        return
        end

```

A6.2.10. Get_Rxn_Data.f

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C GET_RXN_DATA (FORTRAN 77)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
        subroutine get_rxn_data(rxn_rates,num_byproduct,
&    product_id,yields,prod_yield)
#include "params.f"

C
C *** ARGS

```

```

C
    real*8 rxn_rates(*),yields(*),prod_yield
    integer*4 num_byproduct,product_id
C
C ***LOCALS
C
C --> MISC. INDEX
    integer*4 i

C ** COMPUTE THE SUM OF RATES
    sum = 0.0
    do 1000 i=1,num_byproduct
        sum = sum + rxn_rates(i)
1000 continue

C ** COMPUTE THE YIELD
    if(sum .gt. 0.0) then
        do 2000 i=1,num_byproduct
            yields(i) = rxn_rates(i)/sum
2000 continue
C
        prod_yield = rxn_rates(product_id)/sum
    else

        do 2002 i=1,num_byproduct
            yields(i) = -1.0
2002 continue
        prod_yield=-1.0
    endif

    return
end

```

A6.2.11. Update_Ast_List.f

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C UPDATE_AST_LIST (FORTRAN 77)
C
C THIS CODE SAVES THE FEASIBLE STATE TASKS
C "AST" MEANS "AGGREGATE STATE TASK"
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    subroutine update_ast_list
    &          (num_rxn,parent,gnode_list,
    &          gnode_card,byprod_list,byprod_card,
    &          byprod_yields,
    &          child_node_stack,num_child_nodes,
    &          rxn_node_ptr,num_precursor_list,
    &          rxn_byprod,num_byprod_list,rxn_data,
    &          ast_listi,ast_listr,ast_card)
#include "params.f"

C
C *** ARGS
C
    integer*4 num_rxn

C
    integer*4 parent(*)

C
    integer*4 gnode_list(p_max_molecules,*),
    &          gnode_card
C

```

```

integer*4 byprod_list(p_max_molecules,*),
& byprod_card
C
integer*4 child_node_stack(*),
& num_child_nodes
C
integer*4 rxn_node_ptr(p_max_num_rxn,*),
& num_precursor_list(*)
C
integer*4 rxn_byprod(p_max_num_rxn,
& p_max_precursor_per_rxn,*),
& num_byprod_list(p_max_num_rxn,*)
C
real*8 rxn_data(p_max_num_rxn,
& p_max_precursor_per_rxn,*)
C
integer*4 ast_listi(p_max_ast,*), ast_card
C
real*8 ast_listr(p_max_ast,*)
C
real*8 byprod_yields(p_max_num_rxn,
& p_max_precursor_per_rxn,*)
C
C
C *** LOCALS
C
C --> UPDATED LOCATION IN "ast_listi"
integer*4 point_to
C
C --> CURRENT POINTER INTO "child_node_stack"
integer*4 ptr
C
C --> LOCATION OF PARENT IN BYPRODUCT LISTING IN AST MATRIX
integer*4 parent_locate
C
C --> MOLECULE STORAGE SITES AND INCIDENCE FLAG
integer*4 moll(p_num_site),hit
C
C --> IDENTITY OF BYPRODUCT IN BYPRODUCT LIST MATRIX
integer*4 byprod_id
C
C --> SCRATCH STORAGE FOR YIELD
real*8 yield_val
C
C --> MISCELLANEOUS INDEX
integer*4 rxn,pre,i,k

point_to = ast_card
do 1000 rxn=1,num_rxn
  do 1001 pre=1,num_precursor_list(rxn)
    ptr = rxn_node_ptr(rxn,pre)
  end do
end do

C **      RECALL IF THE CURRENT REACTION/PRECURSOR CONSTITUTES A
C      FEASIBLE AST
      if(ptr .ne. p_null_value) then

C **      UPDATE THE AST POINTER
      point_to=point_to+1

C **      ADD RXN IDENTIFIER TO AST DATA

```

```

        ast_listi(point_to,p_ast_rxn_id)= rxn
C **      ADD PRECURSOR ID TO AST DATA
        ast_listi(point_to,p_ast_pre_id)=
&        child_node_stack(ptr)
C **      ADD BYPRODUCT IDS TO AST DATA
        parent_locate=0
        do 1002 i=1,num_byprod_list(rxn,pre)
            byprod_id = rxn_byprod(rxn,pre,i)
            ast_listi(point_to,p_ast_byprod_start+i-1)=
&            byprod_id
C **      CHECK IF ACTUAL PRODUCT FOUND, IF SO MAKE NOTE
            hit=0
            do 2000 k=1,p_num_site
                moll(k) = byprod_list(byprod_id,k)
2000        continue
            call compare_mol(moll,parent,hit)
            if(hit .eq. 1) parent_locate=i
1002        continue
C **      IF PARENT MOLECULE IS NOT IN BYPRODUCT LIST, STOP BECAUSE
C          RESULTS ARE INCORRECT.  OTHERWISE, PLACE THE LOCATION OF
C          PARENT IN AST LIST
        if(parent_locate .eq. 0) then
            write(*,*) '*** STOP ON UNALIGNED BYPRODUCT MATRIX'
            write(*,*) 'rxn ', rxn
            write(*,*) 'parent ',(parent(jj),jj=1,6)
            do kk=1,num_byprod_list(rxn,pre)
                byprod_id=rxn_byprod(rxn,pre,i)
                write(*,*) (byprod_list(byprod_id,jj),jj=1,6)
            end do
            stop
        else
&            ast_listi(point_to,p_ast_par_locate)=
            parent_locate
        endif
C **      STORE THE RATE AND YIELD DATA
        ast_listr(point_to,p_ast_rate_start)=
&        rxn_data(rxn,pre,p_rxn_rate)
        do 3001 i=1,num_byprod_list(rxn,pre)
            yield_val = byprod_yields(rxn,pre,i)
            ast_liscr(point_to,p_ast_yield_start+i-1)=
&            yield_val
C            write(*,*) 'idx : --> ',p_ast_yield_start+i-1,
C            &            yield_val,
C            &            rxn_data(rxn,pre,p_rxn_rate)
3001        continue
C            write(*,*) 'yields',(byprod_yields(rxn,pre,i),
C            &            i=1,num_byprod_list(rxn,pre))
        endif

```

```

1001     continue
1000     continue

C ** UPDATE AST SIZE AND MAKE SURE NO ARRAY BOUNDS ARE VIOLATED
      call update_ast_card(point_to,ast_card)
C
      return
      end

C -----
C CODE FOR UPDATING THE AST CARDINALITY. CHECKS
C TO SEE IF MAXIMUM LEVEL HAS BEEN SUCCEEDED; AND IF SO, TERMINATES
C WITH A STATUS MESSAGE. THE LOCATION OF THIS CODE IS SUCH THAT IT
C IS NOT SUFFICIENT FOR STOPPING EVERY ARRAY VIOLATION BEFORE TERMINATION,
C BUT IF A VIOLATION OCCURS, THIS CODE WILL IDENTIFY IT RETROACTIVELY AND
C THEN TERMINATE. THIS IS DONE FOR COMPUTATIONAL EFFICACY.
      subroutine update_ast_card(point_to,ast_card)
#include "params.f"
      integer*4 point_to,ast_card
C
      ast_card=point_to
      if(ast_card .gt. p_max_ast) then
        write(*,*) '***MAX ASTs EXCEEDED'
        stop
      endif
C
      return
      end

```

A6.2.12. Update_Byprod_List.f

```

C
C subroutine which updates the global node list and returns indices and
C the new cardinality of the list
C
      subroutine update_byprod_list(byprods,num_byproduct,
&                                byprod_indices,byprod_list,byprod_card)
#include "params.f"
C
C *** args
C
      integer*4 byprods(p_max_precursor_per_rxn,*),
&              byprod_list(p_max_molecules,*),
&              byprod_indices(*)
      integer*4 num_byproduct, byprod_card
C
C *** locals
C
C --> PRECURSOR AND GNODE MOLECULES
      integer*4 gmol(p_num_site), pmol(p_num_site)
C --> INDICATES A PRECURSOR ALREADY EXISTS IN "byprod_list"
      integer*4 hit
C --> MISCELLANEOUS INDICES
      integer*4 i,j,k
C --> UPDATED CARDINALITY
      integer*4 new_card
C*****

```

```

C
C
C      new_card = byprod_card
C
C      do 1000 i=1,num_byproduct
C          hit=0
C --> EXTRACT MOLECULES AND COMPARE
C          do 1001 j=1,max(byprod_card,1)
C              do 2000 k=1,p_num_site
C                  pmol(k) = byprods(i,k)
C                  gmol(k) = byprod_list(j,k)
2000          continue
C
C              if ( hit .eq. 0) then
C                  call compare_mol(pmol,gmol,hit)
C --> IF "hit" = 1, STORE THE MOLECULE INDEX
C                  if( hit .eq. 1) then
C                      byprod_indices(i) = j
C                  endif
C              endif
1001          continue
C              if (hit .eq. 0) then
C                  new_card = new_card+1
C                  byprod_indices(i) = new_card
C                  do 3000 k=1,p_num_site
C                      byprod_list(new_card,k) = pmol(k)
3000          continue
C              endif
1000          continue
C
C          call update_byprod(new_card,byprod_card)
C
C          return
C          end

C -----
C CODE FOR UPDATING 'byprod_card' (I.E., INCREMENT BY 1). CHECKS
C TO SEE IF MAXIMUM VALUE HAS BEEN SUCCEEDED; AND IF SO, TERMINATES
C WITH A STATUS MESSAGE
C      subroutine update_byprod(new_card,byprod_card)
#include "params.f"
C      integer*4 new_card,byprod_card
C
C      if(new_card .gt. p_max_molecules)
C          & then
C              write(*,*) '***MAX MOLECULES EXCEEDED, PROGRAM TERMINATED'
C              stop
C          else
C              byprod_card = new_card
C          endif
C
C      return
C      end

```

A6.2.13. Compare_Mol.f

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C COMPARE_MOL (FORTRAN 77)
C compares two polysubstituted benzene molecules and

```

```

C returns "hit=1" if the molecules are identical.  code
C has been vectorized for CRAY-YMP\C90.
C
C coded by and property of jonathan knight
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      subroutine compare_mol(mol1,mol2,hit)
#include "params.f"

C *** args
C molecules
      integer*4 mol1(*), mol2(*)
C flag indicating a hit or miss
      integer*4 hit

C *** locals
      integer*4 store1(2*p_num_site),store2(2*p_num_site),i
      integer*4 d1,d2,d3,d4,d5,d6,norm(p_num_site)

C ** SETUP COMPARISON STRUCTURES FOR FORWARD AND REVERSE ORIENTATIONS
      do 500 i=1,p_num_site
        store1(i) = mol2(i)
        store1(p_num_site+i) = mol2(i)
500    continue
      store2(1) = store1(1)
      store2(p_num_site+1) = store1(p_num_site+1)
      do 600 i=1,p_num_site-1
        store2(i+1) = mol2(7-i)
        store2(i+p_num_site+1) = mol2(p_num_site+1-i)
600    continue
C
C ** INITIALIZE COMPARISON STATUS TO NOT FOUND ("hit=0")
      hit = 0
C
C ** STARTING AT EACH SITE ON THE MOLECULE, COMPARE THE TWO
C STRUCTURES BOTH IN FORWARD AND REVERSE ORIENTATION BY
C TAKING NORMED DIFFERENCE OF THE SUBSTITUENT IDENTIFIERS
      do 1000 i=1,p_num_site

C { would have to be generalized for other substrates!!!}
C
      d1 = mol1(1) - store1(i)
      d2 = mol1(2) - store1(i+1)
      d3 = mol1(3) - store1(i+2)
      d4 = mol1(4) - store1(i+3)
      d5 = mol1(5) - store1(i+4)
      d6 = mol1(6) - store1(i+5)

C
      norm(i) = d1*d1+d2*d2+d3*d3+d4*d4+d5*d5+d6*d6

C
      d1 = mol1(1) - store2(i)
      d2 = mol1(2) - store2(i+1)
      d3 = mol1(3) - store2(i+2)
      d4 = mol1(4) - store2(i+3)
      d5 = mol1(5) - store2(i+4)
      d6 = mol1(6) - store2(i+5)

C
      norm(i) = norm(i)*
& (d1*d1+d2*d2+d3*d3+d4*d4+d5*d5+d6*d6)
C

```



```

C
  1000   continue
C
C ** IF ANY OF THE NORMS ARE ZERO, THAT INDICATES A HIT AND
C   THE MOLECULES MUST BE IDENTICAL

      if(
&      norm(1) .eq. 0 .or.
&      norm(2) .eq. 0 .or.
&      norm(3) .eq. 0 .or.
&      norm(4) .eq. 0 .or.
&      norm(5) .eq. 0 .or.
&      norm(6) .eq. 0 ) then

          hit = 1
          return
      endif
C
      return
      end

```

A6.2.14. Compare_Mol2.f

```

C-----
C subroutine to compare two molecules <-- subject for further optimization.
C in this routine, an array is incorporated which specifies the substructure
C to be compared. Substructure reference is wrt mol1.
C   for example, mol1 = <1 0 2 0 0 0>
C                   mol2 = <4 1 3 2 1 1>
C   and template = <1 3> then
C       substructure = <0 1 0 3 0 0>
C
C the output is each unique substructure found and the total number found.

      subroutine compare_mol2(mol1,mol2,template,num_groups,
& substructure, num_hit)
#include "params.f"
C *** args
C molecules
      integer*4 mol1(*), mol2(*)
C substructure template and its cardinality (wrt mol1)
      integer*4 template(*),num_groups
C location of identical substructures and their cardinality (mapped
C from template)
      integer*4 substructure(p_max_precursor_per_rxn,*),num_hit

C *** locals
      integer*4 store1(2*p_num_site),store2(2*p_num_site),i,j
      integer*4 d,norm1,norm2

C *****
C ** construct arrays for forward and backward comparison
      do 500 i=1,p_num_site
          store1(i) = mol2(i)
          store1(p_num_site+i) = mol2(i)
500      continue
      store2(1) = store1(1)
      store2(7) = store1(7)
      do 600 i=1,p_num_site-1
          store2(i+1) = mol2(7-i)
          store2(i+p_num_site+1) = mol2(p_num_site+1-i)
600      continue

```

```

600     continue
C
      num_hit = 0
C
C ** parse through store arrays comparing substructures. compute
C norm of difference in substituent values.

      do 1000 i=1,p_num_site
C
      norm1 = 0
      do 1001 j=1,num_groups
        index = template(j)
        d = moll(index) - store1(i-1+index)
        norm1 = norm1 + d*d
1001     continue

C ** if a hit is achieved, map out the template array in the
C substructure array to indicate where moll substructure falls
C inside mol2
C
      if( norm1 .eq. 0 ) then
        num_hit = num_hit + 1
        do 1002 j=1,num_groups
          index = template(j)
          index2 = i-1 + index

          if ( index2 .le. p_num_site) then
            substructure(num_hit,index2) = index
          else
            index3 = i-(p_num_site+1) + index
            substructure(num_hit,index3) = index
          endif
1002     continue

        endif

      if (num_groups .gt. 1) then
        norm2 = 0
        do 1004 j=1,num_groups
          index = template(j)
          d = moll(index) - store2(i-1+index)
          norm2 = norm2 + d*d
1004     continue
C
      if( norm2 .eq. 0 ) then
        num_hit = num_hit + 1
        do 1003 j=1,num_groups
          index = template(j)
          index2 = i-1 + index

          if ( index2 .eq. 1 .or. index2 .eq.
& p_num_site+1) then
            substructure(num_hit,1) = index
          elseif ( index2 .le. p_num_site) then
            substructure(num_hit,p_num_site+2
& -index2) = index
          else
            index3 = i-(p_num_site+1) + index
            substructure(num_hit,p_num_site+2-index3)
& = index
          endif
      endif

```

```
1003         continue

           endif
           endif
C
1000 continue
C
           return
           end
```

A6.2.15. Params.f

```
C maximum number of global iterations
   integer*4 p_max_it
   parameter(p_max_it=10000000)

C maximum fathomed level
   integer*4 p_max_level
   parameter(p_max_level=5)

C number of substitution sites
   integer*4 p_num_site
   parameter(p_num_site=6)

C maximum number of different reactions
   integer*4 p_max_num_rxn
   parameter(p_max_num_rxn=40)

C maximum number of nodes per level
   integer*4 p_max_nodes_per_level
   parameter(p_max_nodes_per_level=120)

C upper limit on total number of molecules
   integer*4 p_max_molecules
   parameter(p_max_molecules=5000)

C maximum number of precursors per reaction
   integer*4 p_max_precursor_per_rxn
   parameter(p_max_precursor_per_rxn=40)

C generic null value parameter
   integer*4 p_null_value
   parameter(p_null_value=-32676)

C number of components to the reaction data
   integer*4 p_max_rxn_data
   parameter(p_max_rxn_data=2)

C location of yield data in reaction data matrix
   integer*4 p_rxn_yield
   parameter(p_rxn_yield=2)

C location of rate data in reaction data matrix
   integer*4 p_rxn_rate
   parameter(p_rxn_rate=1)

C maximum number of aggregate state tasks
   integer*4 p_max_ast
   parameter(p_max_ast=20000)

C precursor id location
```

```
integer*4 p_ast_pre_id
parameter(p_ast_pre_id=1)

C rxn id location
integer*4 p_ast_rxn_id
parameter(p_ast_rxn_id=2)

C parent locator location
integer*4 p_ast_par_locate
parameter(p_ast_par_locate=3)

C starting point for byproduct identifiers in ast_datai
integer*4 p_ast_byprod_start
parameter(p_ast_byprod_start=4)

C starting point rate data in ast_datar
integer*4 p_ast_rate_start
parameter(p_ast_rate_start=1)

C starting point yield data in ast_datar
integer*4 p_ast_yield_start
parameter(p_ast_yield_start=p_ast_rate_start+1)

C number of integer data locations
integer*4 p_num_ast_datai
parameter(p_num_ast_datai=p_ast_byprod_start+
& p_max_precursor_per_rxn-1)

C number of real data locations
integer*4 p_num_ast_datar
parameter(p_num_ast_datar=p_ast_yield_start+
& p_max_precursor_per_rxn-1)

C minimum acceptable yield in any single synthetic step
real*8 p_min_step_yield
parameter(p_min_step_yield = 0.0)

C minimum acceptable rate in any single synthetic step
real*8 p_min_step_rate
parameter(p_min_step_rate = 0.0)

C minimum acceptable yield in a synthetic pathway
real*8 p_min_path_yield
parameter(p_min_path_yield = 0.0)

C maximum number of types of substituents
integer*4 p_max_substit
parameter(p_max_substit=20)

C total number of reaction sets
integer*4 p_tot_num_rxn
parameter(p_tot_num_rxn=1)

C beginning of list of number of reactions
C in the set
integer*4 p_rxn_set_size_start
parameter(p_rxn_set_size_start=
& p_tot_num_rxn+1)
```

A6.3. Production Runs

Production runs from chapter eight are included in this section. The individual substituent types are identified by numerical assignments given in Table A7.1.

Table A6.1. *Substituent Identification Numbers in the Generated Compounds.*

substituent type	identification number
-H	1
-Cl	2
-CN	3
-NO ₂	4
-NH ₂	5

Thus, the molecule structures are defined by a string of six numbers. An example is given in Figure 1.

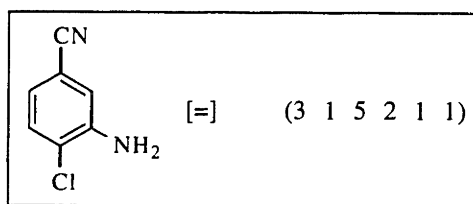


Figure A6.1. *Correspondence Between Molecule Specification and Structure.*

A6.3.1. Run #1: 3-Amino-4-Chlorobenzonitrile

---	CHEMICAL CONVERSION	1						
	PRECURSOR:	3	1	4	2	1	1	
	*BYPRODUCT:	3	1	5	2	1	1	YIELD = 1.00000 RATE = 1.00000

---	CHEMICAL CONVERSION	2						
	PRECURSOR:	5	1	4	2	1	1	
	*BYPRODUCT:	3	1	4	2	1	1	YIELD = 0.85714 RATE = 1.00000
	BYPRODUCT:	9	1	4	2	1	1	YIELD = 0.14286

---	CHEMICAL CONVERSION	3						
	PRECURSOR:	3	1	2	1	1		
	BYPRODUCT:	3	4	1	2	1	1	YIELD = 0.00592
	*BYPRODUCT:	3	1	4	2	1	1	YIELD = 0.99408 RATE = 0.00003

---	CHEMICAL CONVERSION	4						
	PRECURSOR:	3	5	4	2	1	1	
	*BYPRODUCT:	3	1	4	2	1	1	YIELD = 1.00000 RATE = 1.00000

---	CHEMICAL CONVERSION	5						

PRECURSOR:	3	1	4	2	5	1			
*BYPRODUCT:	3	1	4	2	1	1	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION	6								
PRECURSOR:	3	1	4	2	1	5			
*BYPRODUCT:	3	1	4	2	1	1	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION	7								
PRECURSOR:	5	1	4	1	1	1			
BYPRODUCT:	5	2	4	1	1	1	YIELD =	0.18951	
*BYPRODUCT:	5	1	4	2	1	1	YIELD =	0.77669	RATE = 0.01570
BYPRODUCT:	5	1	4	1	2	1	YIELD =	0.00000	
BYPRODUCT:	4	1	5	2	1	1	YIELD =	0.03380	

--- CHEMICAL CONVERSION	8								
PRECURSOR:	5	1	1	2	1	1			
*BYPRODUCT:	3	1	1	2	1	1	YIELD =	0.85714	RATE = 1.00000
BYPRODUCT:	9	1	1	2	1	1	YIELD =	0.14286	

--- CHEMICAL CONVERSION	9								
PRECURSOR:	3	5	1	2	1	1			
*BYPRODUCT:	3	1	1	2	1	1	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION	10								
PRECURSOR:	3	1	5	2	1	1			
*BYPRODUCT:	3	1	1	2	1	1	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION	11								
PRECURSOR:	4	1	1	2	1	1			
*BYPRODUCT:	5	1	1	2	1	1	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION	12								
PRECURSOR:	5	1	1	1	1	1			
BYPRODUCT:	5	2	1	1	1	1	YIELD =	0.32796	
BYPRODUCT:	5	1	2	1	1	1	YIELD =	0.00000	
*BYPRODUCT:	5	1	1	2	1	1	YIELD =	0.67204	RATE =
2500000.00000									

--- CHEMICAL CONVERSION	13								
PRECURSOR:	1	1	1	2	1	1			
*BYPRODUCT:	4	1	1	2	1	1	YIELD =	0.67821	RATE = 0.13000
BYPRODUCT:	1	4	1	2	1	1	YIELD =	0.00876	
BYPRODUCT:	1	1	4	2	1	1	YIELD =	0.31302	

--- CHEMICAL CONVERSION	14								
PRECURSOR:	4	5	1	2	1	1			
*BYPRODUCT:	4	1	1	2	1	1	YIELD =	1.00000	RATE = 1.00000

```

-----
--- CHEMICAL CONVERSION      15
-----
PRECURSOR:      4   1   5   2   1   1
*BYPRODUCT:    4   1   1   2   1   1   YIELD = 1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      16
-----
PRECURSOR:      4   1   1   1   1   1
*BYPRODUCT:    5   1   1   1   1   1   YIELD = 1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      17
-----
PRECURSOR:      3   4   1   2   1   1
*BYPRODUCT:    3   5   1   2   1   1   YIELD = 1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      18
-----
PRECURSOR:      5   4   1   2   1   1
*BYPRODUCT:    3   4   1   2   1   1   YIELD = 0.85714   RATE = 1.00000
BYPRODUCT:     9   4   1   2   1   1   YIELD = 0.14286

-----
--- CHEMICAL CONVERSION      19
-----
PRECURSOR:      3   4   5   2   1   1
*BYPRODUCT:    3   4   1   2   1   1   YIELD = 1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      20
-----
PRECURSOR:      3   4   1   2   5   1
*BYPRODUCT:    3   4   1   2   1   1   YIELD = 1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      21
-----
PRECURSOR:      3   4   1   2   1   5
*BYPRODUCT:    3   4   1   2   1   1   YIELD = 1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      22
-----
PRECURSOR:      3   1   4   2   1   1
*BYPRODUCT:    3   1   5   2   1   1   YIELD = 1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      23
-----
PRECURSOR:      5   1   4   2   1   1
*BYPRODUCT:    3   1   4   2   1   1   YIELD = 0.85714   RATE = 1.00000
BYPRODUCT:     9   1   4   2   1   1   YIELD = 0.14286

-----
--- CHEMICAL CONVERSION      24
-----
PRECURSOR:      3   1   1   2   1   1
BYPRODUCT:     3   4   1   2   1   1   YIELD = 0.00592
*BYPRODUCT:    3   1   4   2   1   1   YIELD = 0.99408   RATE = 0.00003

-----
--- CHEMICAL CONVERSION      25
-----

```

```

PRECURSOR:      3  5  4  2  1  1
*BYPRODUCT:    3  1  4  2  1  1  YIELD =  1.00000  RATE =  1.00000

```

```

-----
--- CHEMICAL CONVERSION      26
-----

```

```

PRECURSOR:      3  1  4  2  5  1
*BYPRODUCT:    3  1  4  2  1  1  YIELD =  1.00000  RATE =  1.00000

```

```

-----
--- CHEMICAL CONVERSION      27
-----

```

```

PRECURSOR:      3  1  4  2  1  5
*BYPRODUCT:    3  1  4  2  1  1  YIELD =  1.00000  RATE =  1.00000

```

```

-----
--- CHEMICAL CONVERSION      29
-----

```

```

PRECURSOR:      3  5  1  2  1  1
*BYPRODUCT:    3  5  4  2  1  1  YIELD =  0.62961  RATE =  0.11669
BYPRODUCT:     3  5  1  2  4  1  YIELD =  0.37039
BYPRODUCT:     3  5  1  2  1  4  YIELD =  0.00000

```

```

-----
--- CHEMICAL CONVERSION      29
-----

```

```

PRECURSOR:      3  2  4  2  1  1
*BYPRODUCT:    3  5  4  2  1  1  YIELD =  0.50000  RATE =  1.00000
BYPRODUCT:     3  2  4  5  1  1  YIELD =  0.50000

```

```

-----
--- CHEMICAL CONVERSION      30
-----

```

```

PRECURSOR:      3  4  1  2  1  1
*BYPRODUCT:    3  5  1  2  1  1  YIELD =  1.00000  RATE =  1.00000

```

A6.3.2. Run #2: 2-Amino-4-Chlorobenzonitrile

```

-----
--- CHEMICAL CONVERSION      1
-----

```

```

PRECURSOR:      3  4  1  2  1  1
*BYPRODUCT:    3  5  1  2  1  1  YIELD =  1.00000  RATE =  1.00000

```

```

-----
--- CHEMICAL CONVERSION      2
-----

```

```

PRECURSOR:      5  4  1  2  1  1
*BYPRODUCT:    3  4  1  2  1  1  YIELD =  0.85714  RATE =  1.00000
BYPRODUCT:     9  4  1  2  1  1  YIELD =  0.14286

```

```

-----
--- CHEMICAL CONVERSION      3
-----

```

```

PRECURSOR:      3  4  5  2  1  1
*BYPRODUCT:    3  4  1  2  1  1  YIELD =  1.00000  RATE =  1.00000

```

```

-----
--- CHEMICAL CONVERSION      4
-----

```

```

PRECURSOR:      3  4  1  2  5  1
*BYPRODUCT:    3  4  1  2  1  1  YIELD =  1.00000  RATE =  1.00000

```

```

-----
--- CHEMICAL CONVERSION      5
-----

```

	PRECURSOR:	3	4	1	2	1	5			
	*BYPRODUCT:	3	4	1	2	1	1	YIELD =	1.00000	RATE = 1.00000

	---	CHEMICAL CONVERSION	6							

	PRECURSOR:	5	1	1	2	1	1			
	*BYPRODUCT:	5	4	1	2	1	1	YIELD =	0.99868	RATE =
14.98728	BYPRODUCT:	5	1	4	2	1	1	YIELD =	0.00132	

	---	CHEMICAL CONVERSION	7							

	PRECURSOR:	2	4	1	2	1	1			
	*BYPRODUCT:	5	4	1	2	1	1	YIELD =	1.00000	RATE = 1.00000
	BYPRODUCT:	5	1	4	2	1	1	YIELD =	0.00000	

	---	CHEMICAL CONVERSION	8							

	PRECURSOR:	5	4	1	1	1	1			
	BYPRODUCT:	5	4	2	1	1	1	YIELD =	0.00000	
	*BYPRODUCT:	5	4	1	2	1	1	YIELD =	0.80386	RATE = 0.07575
	BYPRODUCT:	4	5	1	2	1	1	YIELD =	0.00000	
	BYPRODUCT:	5	4	1	1	1	2	YIELD =	0.19614	
	BYPRODUCT:	5	4	1	2	1	2	YIELD =	0.00000	

	---	CHEMICAL CONVERSION	9							

	PRECURSOR:	4	1	1	2	1	1			
	*BYPRODUCT:	5	1	1	2	1	1	YIELD =	1.00000	RATE = 1.00000

	---	CHEMICAL CONVERSION	10							

	PRECURSOR:	5	1	1	1	1	1			
	BYPRODUCT:	5	2	1	1	1	1	YIELD =	0.32796	
	BYPRODUCT:	5	1	2	1	1	1	YIELD =	0.00000	
2500000.00000	*BYPRODUCT:	5	1	1	2	1	1	YIELD =	0.67204	RATE =

	---	CHEMICAL CONVERSION	11							

	PRECURSOR:	1	1	1	2	1	1			
	*BYPRODUCT:	4	1	1	2	1	1	YIELD =	0.67821	RATE = 0.13000
	BYPRODUCT:	1	4	1	2	1	1	YIELD =	0.00876	
	BYPRODUCT:	1	1	4	2	1	1	YIELD =	0.31302	

	---	CHEMICAL CONVERSION	12							

	PRECURSOR:	4	5	1	2	1	1			
	*BYPRODUCT:	4	1	1	2	1	1	YIELD =	1.00000	RATE = 1.00000

	---	CHEMICAL CONVERSION	13							

	PRECURSOR:	4	1	5	2	1	1			
	*BYPRODUCT:	4	1	1	2	1	1	YIELD =	1.00000	RATE = 1.00000

	---	CHEMICAL CONVERSION	14							

PRECURSOR:	4	1	1	1	1	1			
*BYPRODUCT:	5	1	1	1	1	1	YIELD =	1.00000	RATE = 1.00000

---	CHEMICAL CONVERSION	15							

PRECURSOR:	2	1	1	2	1	1			
*BYPRODUCT:	2	4	1	2	1	1	YIELD =	1.00000	RATE = 0.00010

---	CHEMICAL CONVERSION	16							

PRECURSOR:	1	1	1	2	1	1			
*BYPRODUCT:	2	1	1	2	1	1	YIELD =	0.62459	RATE = 0.06540
BYPRODUCT:	1	2	1	2	1	1	YIELD =	0.01249	
BYPRODUCT:	1	1	2	2	1	1	YIELD =	0.36291	

---	CHEMICAL CONVERSION	17							

PRECURSOR:	2	5	1	2	1	1			
*BYPRODUCT:	2	1	1	2	1	1	YIELD =	1.00000	RATE = 1.00000

---	CHEMICAL CONVERSION	18							

PRECURSOR:	5	1	1	1	1	1			
*BYPRODUCT:	5	4	1	1	1	1	YIELD =	0.77269	RATE =
17842.00000	BYPRODUCT:	5	1	4	1	1	YIELD =	0.00003	
BYPRODUCT:	5	1	1	4	1	1	YIELD =	0.22728	

---	CHEMICAL CONVERSION	19							

PRECURSOR:	2	4	1	1	1	1			
*BYPRODUCT:	5	4	1	1	1	1	YIELD =	1.00000	RATE = 1.00000

---	CHEMICAL CONVERSION	20							

PRECURSOR:	1	1	1	2	1	1			
*BYPRODUCT:	1	1	4	2	1	1	YIELD =	0.31302	RATE = 0.06000
BYPRODUCT:	1	4	1	2	1	1	YIELD =	0.00876	
BYPRODUCT:	4	1	1	2	1	1	YIELD =	0.67821	

---	CHEMICAL CONVERSION	21							

PRECURSOR:	2	4	5	1	1	1			
*BYPRODUCT:	1	1	4	2	1	1	YIELD =	1.00000	RATE = 1.00000

---	CHEMICAL CONVERSION	22							

PRECURSOR:	2	4	1	5	1	1			
*BYPRODUCT:	1	1	4	2	1	1	YIELD =	1.00000	RATE = 1.00000

---	CHEMICAL CONVERSION	23							

PRECURSOR:	2	4	1	1	5	1			
*BYPRODUCT:	1	1	4	2	1	1	YIELD =	1.00000	RATE = 1.00000

---	CHEMICAL CONVERSION	24							

```

-----
  PRECURSOR:      2  4  1  1  1  5
  *BYPRODUCT:    1  1  4  2  1  1  YIELD = 1.00000  RATE = 1.00000
-----

--- CHEMICAL CONVERSION 25
-----
  PRECURSOR:      3  1  5  2  1  1
  *BYPRODUCT:    3  4  5  2  1  1  YIELD = 0.62716  RATE = 0.00070
  BYPRODUCT:     3  1  5  2  4  1  YIELD = 0.00389
  BYPRODUCT:     3  1  5  2  1  4  YIELD = 0.36894
-----

--- CHEMICAL CONVERSION 26
-----
  PRECURSOR:      3  4  2  2  1  1
  *BYPRODUCT:    3  4  5  2  1  1  YIELD = 1.00000  RATE = 1.00000
  BYPRODUCT:     3  4  2  5  1  1  YIELD = 0.00000
-----

--- CHEMICAL CONVERSION 27
-----
  PRECURSOR:      3  1  4  2  1  1
  *BYPRODUCT:    3  1  5  2  1  1  YIELD = 1.00000  RATE = 1.00000
-----

--- CHEMICAL CONVERSION 28
-----
  PRECURSOR:      2  4  1  5  1  1
  *BYPRODUCT:    3  1  4  2  1  1  YIELD = 0.85714  RATE = 1.00000
  BYPRODUCT:     9  1  4  2  1  1  YIELD = 0.14286
-----

--- CHEMICAL CONVERSION 29
-----
  PRECURSOR:      3  1  1  2  1  1
  BYPRODUCT:     3  4  1  2  1  1  YIELD = 0.00592
  *BYPRODUCT:    3  1  4  2  1  1  YIELD = 0.99408  RATE = 0.00003
-----

--- CHEMICAL CONVERSION 30
-----
  PRECURSOR:      3  5  4  2  1  1
  *BYPRODUCT:    3  1  4  2  1  1  YIELD = 1.00000  RATE = 1.00000
-----

--- CHEMICAL CONVERSION 31
-----
  PRECURSOR:      3  1  4  2  5  1
  *BYPRODUCT:    3  1  4  2  1  1  YIELD = 1.00000  RATE = 1.00000
-----

--- CHEMICAL CONVERSION 32
-----
  PRECURSOR:      3  1  4  2  1  5
  *BYPRODUCT:    3  1  4  2  1  1  YIELD = 1.00000  RATE = 1.00000
-----

--- CHEMICAL CONVERSION 33
-----
  PRECURSOR:      3  1  5  2  1  1
  *BYPRODUCT:    3  1  5  2  1  4  YIELD = 0.36894  RATE = 0.00041
  BYPRODUCT:     3  1  5  2  4  1  YIELD = 0.00389
  BYPRODUCT:     3  4  5  2  1  1  YIELD = 0.62716
-----

```

```

-----
--- CHEMICAL CONVERSION      34
-----
PRECURSOR:      3  4  1  2  2  1
BYPRODUCT:      3  4  1  5  2  1  YIELD =  0.00000
*BYPRODUCT:     3  1  5  2  1  4  YIELD =  1.00000      RATE = 1.00000

```

```

-----
--- CHEMICAL CONVERSION      35
-----
PRECURSOR:      3  1  4  2  1  1
*BYPRODUCT:     3  1  5  2  1  1  YIELD =  1.00000      RATE = 1.00000

```

A6.3.3. Run #3: 2-Aminobenzonitrile

```

-----
--- CHEMICAL CONVERSION      1
-----
PRECURSOR:      3  4  1  1  1  1
*BYPRODUCT:     3  5  1  1  1  1  YIELD =  1.00000      RATE = 1.00000

```

```

-----
--- CHEMICAL CONVERSION      2
-----
PRECURSOR:      5  4  1  1  1  1
*BYPRODUCT:     3  4  1  1  1  1  YIELD =  0.85714      RATE = 1.00000
BYPRODUCT:      9  4  1  1  1  1  YIELD =  0.14286

```

```

-----
--- CHEMICAL CONVERSION      3
-----
PRECURSOR:      3  1  1  1  1  1
*BYPRODUCT:     3  4  1  1  1  1  YIELD =  0.17200      RATE = 0.00019
BYPRODUCT:      3  1  4  1  1  1  YIELD =  0.80808
BYPRODUCT:      3  1  1  4  1  1  YIELD =  0.01992

```

```

-----
--- CHEMICAL CONVERSION      4
-----
PRECURSOR:      3  4  5  1  1  1
*BYPRODUCT:     3  4  1  1  1  1  YIELD =  1.00000      RATE = 1.00000

```

```

-----
--- CHEMICAL CONVERSION      5
-----
PRECURSOR:      3  4  1  5  1  1
*BYPRODUCT:     3  4  1  1  1  1  YIELD =  1.00000      RATE = 1.00000

```

```

-----
--- CHEMICAL CONVERSION      6
-----
PRECURSOR:      3  4  1  1  5  1
*BYPRODUCT:     3  4  1  1  1  1  YIELD =  1.00000      RATE = 1.00000

```

```

-----
--- CHEMICAL CONVERSION      7
-----
PRECURSOR:      3  4  1  1  1  5
*BYPRODUCT:     3  4  1  1  1  1  YIELD =  1.00000      RATE = 1.00000

```

```

-----
--- CHEMICAL CONVERSION      8
-----
PRECURSOR:      5  1  1  1  1  1
*BYPRODUCT:     5  4  1  1  1  1  YIELD =  0.77269      RATE =
17842.00000
BYPRODUCT:      5  1  4  1  1  1  YIELD =  0.00003
BYPRODUCT:      5  1  1  4  1  1  YIELD =  0.22726

```

```

-----
--- CHEMICAL CONVERSION      9
-----
  PRECURSOR:      2  4  1  1  1  1
  *BYPRODUCT:     5  4  1  1  1  1   YIELD =  1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION     10
-----
  PRECURSOR:      4  1  1  1  1  1
  *BYPRODUCT:     5  1  1  1  1  1   YIELD =  1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION     11
-----
  PRECURSOR:      1  1  1  1  1  1
  *BYPRODUCT:     4  1  1  1  1  1   YIELD =  1.00000   RATE = 6.00000

-----
--- CHEMICAL CONVERSION     12
-----
  PRECURSOR:      5  4  1  1  1  1
  *BYPRODUCT:     4  1  1  1  1  1   YIELD =  1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION     13
-----
  PRECURSOR:      4  1  5  1  1  1
  *BYPRODUCT:     4  1  1  1  1  1   YIELD =  1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION     14
-----
  PRECURSOR:      4  1  1  5  1  1
  *BYPRODUCT:     4  1  1  1  1  1   YIELD =  1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION     15
-----
  PRECURSOR:      2  1  1  1  1  1
  *BYPRODUCT:     2  4  1  1  1  1   YIELD =  0.31302   RATE = 0.06000
  BYPRODUCT:      2  1  4  1  1  1   YIELD =  0.00876
  BYPRODUCT:      2  1  1  4  1  1   YIELD =  0.67821

-----
--- CHEMICAL CONVERSION     16
-----
  PRECURSOR:      2  4  5  1  1  1
  *BYPRODUCT:     2  4  1  1  1  1   YIELD =  1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION     17
-----
  PRECURSOR:      2  4  1  5  1  1
  *BYPRODUCT:     2  4  1  1  1  1   YIELD =  1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION     18
-----
  PRECURSOR:      2  4  1  1  5  1
  *BYPRODUCT:     2  4  1  1  1  1   YIELD =  1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION     19
-----

```

PRECURSOR:	2	4	1	1	1	5			
*BYPRODUCT:	2	4	1	1	1	1	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION						20			

PRECURSOR:	1	1	1	1	1	1			
*BYPRODUCT:	2	1	1	1	1	1	YIELD =	1.00000	RATE = 6.00000

--- CHEMICAL CONVERSION						21			

PRECURSOR:	2	5	1	1	1	1			
*BYPRODUCT:	2	1	1	1	1	1	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION						22			

PRECURSOR:	2	1	5	1	1	1			
*BYPRODUCT:	2	1	1	1	1	1	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION						23			

PRECURSOR:	2	1	1	5	1	1			
*BYPRODUCT:	2	1	1	1	1	1	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION						24			

PRECURSOR:	2	1	5	1	1	1			
*BYPRODUCT:	2	4	5	1	1	1	YIELD =	0.16887	RATE =
267.63000									
BYPRODUCT:	2	1	5	4	1	1	YIELD =	0.73178	
BYPRODUCT:	2	1	5	1	4	1	YIELD =	0.00000	
BYPRODUCT:	2	1	5	1	1	4	YIELD =	0.02934	

--- CHEMICAL CONVERSION						25			

PRECURSOR:	2	4	2	1	1	1			
*BYPRODUCT:	2	4	5	1	1	1	YIELD =	1.00000	RATE = 2.00000

--- CHEMICAL CONVERSION						26			

PRECURSOR:	4	1	5	1	1	1			
*BYPRODUCT:	2	4	1	5	1	1	YIELD =	0.77669	RATE = 0.01570
BYPRODUCT:	2	4	1	1	1	5	YIELD =	0.18951	
BYPRODUCT:	2	5	1	4	1	1	YIELD =	0.03360	
BYPRODUCT:	2	1	5	1	4	1	YIELD =	0.00000	

--- CHEMICAL CONVERSION						27			

PRECURSOR:	2	1	5	1	1	1			
*BYPRODUCT:	2	1	5	1	1	4	YIELD =	0.09934	RATE =
157.44000									
BYPRODUCT:	2	1	5	1	4	1	YIELD =	0.00000	
BYPRODUCT:	2	1	5	4	1	1	YIELD =	0.73178	
BYPRODUCT:	2	4	5	1	1	1	YIELD =	0.16887	

--- CHEMICAL CONVERSION						28			

PRECURSOR:	2	4	1	1	2	1			
BYPRODUCT:	2	1	5	4	1	1	YIELD =	0.50000	

```

*BYPRODUCT:      2  1  5  1  1  4  YIELD =  0.50000  RATE = 1.00000

-----
--- CHEMICAL CONVERSION  29
-----
  PRECURSOR:      4  1  5  1  1  1
  *BYPRODUCT:     2  4  1  1  1  5  YIELD =  0.18951  RATE = 0.00383
  BYPRODUCT:      2  4  1  5  1  1  YIELD =  0.77669
  BYPRODUCT:      2  1  5  1  4  1  YIELD =  0.00000
  BYPRODUCT:      2  5  1  4  1  1  YIELD =  0.03380

-----
--- CHEMICAL CONVERSION  30
-----
  PRECURSOR:      5  1  1  1  1  1
  *BYPRODUCT:     3  1  1  1  1  1  YIELD =  0.85714  RATE = 1.00000
  BYPRODUCT:      9  1  1  1  1  1  YIELD =  0.14286

-----
--- CHEMICAL CONVERSION  31
-----
  PRECURSOR:      3  5  1  1  1  1
  *BYPRODUCT:     3  1  1  1  1  1  YIELD =  1.00000  RATE = 1.00000

-----
--- CHEMICAL CONVERSION  32
-----
  PRECURSOR:      3  1  5  1  1  1
  *BYPRODUCT:     3  1  1  1  1  1  YIELD =  1.00000  RATE = 1.00000

-----
--- CHEMICAL CONVERSION  33
-----
  PRECURSOR:      3  1  1  5  1  1
  *BYPRODUCT:     3  1  1  1  1  1  YIELD =  1.00000  RATE = 1.00000

-----
--- CHEMICAL CONVERSION  34
-----
  PRECURSOR:      4  1  1  1  1  1
  *BYPRODUCT:     5  1  1  1  1  1  YIELD =  1.00000  RATE = 1.00000

-----
--- CHEMICAL CONVERSION  35
-----
  PRECURSOR:      3  4  1  1  1  1
  *BYPRODUCT:     3  5  1  1  1  1  YIELD =  1.00000  RATE = 1.00000

-----
--- CHEMICAL CONVERSION  36
-----
  PRECURSOR:      5  4  1  1  1  1
  *BYPRODUCT:     3  4  1  1  1  1  YIELD =  0.85714  RATE = 1.00000
  BYPRODUCT:      9  4  1  1  1  1  YIELD =  0.14286

-----
--- CHEMICAL CONVERSION  37
-----
  PRECURSOR:      3  1  1  1  1  1
  *BYPRODUCT:     3  4  1  1  1  1  YIELD =  0.17200  RATE = 0.00019
  BYPRODUCT:      3  1  4  1  1  1  YIELD =  0.80808
  BYPRODUCT:      3  1  1  4  1  1  YIELD =  0.01992

-----
--- CHEMICAL CONVERSION  38
-----

```

PRECURSOR:	3	4	5	1	1	1			
*BYPRODUCT:	3	4	1	1	1	1	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION									39

PRECURSOR:	3	4	1	5	1	1			
*BYPRODUCT:	3	4	1	1	1	1	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION									40

PRECURSOR:	3	4	1	1	5	1			
*BYPRODUCT:	3	4	1	1	1	1	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION									41

PRECURSOR:	3	4	1	1	1	5			
*BYPRODUCT:	3	4	1	1	1	1	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION									42

PRECURSOR:	3	1	4	1	1	1			
*BYPRODUCT:	3	1	5	1	1	1	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION									43

PRECURSOR:	4	1	5	1	1	1			
*BYPRODUCT:	3	1	4	1	1	1	YIELD =	0.85714	RATE = 1.00000
BYPRODUCT:	9	1	4	1	1	1	YIELD =	0.14286	

--- CHEMICAL CONVERSION									44

PRECURSOR:	3	1	1	1	1	1			
BYPRODUCT:	3	4	1	1	1	1	YIELD =	0.17200	
*BYPRODUCT:	3	1	4	1	1	1	YIELD =	0.80808	RATE = 0.00087
BYPRODUCT:	3	1	1	4	1	1	YIELD =	0.01992	

--- CHEMICAL CONVERSION									45

PRECURSOR:	3	5	4	1	1	1			
*BYPRODUCT:	3	1	4	1	1	1	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION									46

PRECURSOR:	3	1	4	5	1	1			
*BYPRODUCT:	3	1	4	1	1	1	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION									47

PRECURSOR:	3	1	4	1	5	1			
*BYPRODUCT:	3	1	4	1	1	1	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION									48

PRECURSOR:	3	1	4	1	1	5			
*BYPRODUCT:	3	1	4	1	1	1	YIELD =	1.00000	RATE = 1.00000


```

-----
--- CHEMICAL CONVERSION      49
-----
  PRECURSOR:      3   1   1   4   1   1
  *BYPRODUCT:     3   1   1   5   1   1   YIELD =   1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      50
-----
  PRECURSOR:      4   1   1   5   1   1
  *BYPRODUCT:     3   1   1   4   1   1   YIELD =   0.85714   RATE = 1.00000
  BYPRODUCT:      9   1   1   4   1   1   YIELD =   0.14286

-----
--- CHEMICAL CONVERSION      51
-----
  PRECURSOR:      3   5   1   4   1   1
  *BYPRODUCT:     3   1   1   4   1   1   YIELD =   1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      52
-----
  PRECURSOR:      3   1   5   4   1   1
  *BYPRODUCT:     3   1   1   4   1   1   YIELD =   1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      53
-----
  PRECURSOR:      3   1   5   1   1   1
  *BYPRODUCT:     3   4   5   1   1   1   YIELD =   0.54941   RATE = 0.82787
  BYPRODUCT:      3   1   5   4   1   1   YIELD =   0.12729
  BYPRODUCT:      3   1   5   1   4   1   YIELD =   0.00010
  BYPRODUCT:      3   1   5   1   1   4   YIELD =   0.32320

-----
--- CHEMICAL CONVERSION      54
-----
  PRECURSOR:      3   4   2   1   1   1
  *BYPRODUCT:     3   4   5   1   1   1   YIELD =   1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      55
-----
  PRECURSOR:      3   1   4   1   1   1
  *BYPRODUCT:     3   1   5   1   1   1   YIELD =   1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      56
-----
  PRECURSOR:      2   4   5   1   1   1
  *BYPRODUCT:     3   4   2   1   1   1   YIELD =   0.85714   RATE = 1.00000
  BYPRODUCT:      9   4   2   1   1   1   YIELD =   0.14286

-----
--- CHEMICAL CONVERSION      57
-----
  PRECURSOR:      3   4   2   5   1   1
  *BYPRODUCT:     3   4   2   1   1   1   YIELD =   1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      58
-----
  PRECURSOR:      3   4   2   1   5   1
  *BYPRODUCT:     3   4   2   1   1   1   YIELD =   1.00000   RATE = 1.00000

```

```

-----
--- CHEMICAL CONVERSION      59
-----
  PRECURSOR:      3   4   2   1   1   5
  *BYPRODUCT:     3   4   2   1   1   1   YIELD =  1.00000   RATE =  1.00000

-----
--- CHEMICAL CONVERSION      60
-----
  PRECURSOR:      2   1   5   1   1   1
  *BYPRODUCT:     2   4   5   1   1   1   YIELD =  0.16887   RATE =
267.63000
  BYPRODUCT:      2   1   5   4   1   1   YIELD =  0.73178
  BYPRODUCT:      2   1   5   1   4   1   YIELD =  0.00000
  BYPRODUCT:      2   1   5   1   1   4   YIELD =  0.09934

-----
--- CHEMICAL CONVERSION      61
-----
  PRECURSOR:      2   4   2   1   1   1
  *BYPRODUCT:     2   4   5   1   1   1   YIELD =  1.00000   RATE =  2.00000

-----
--- CHEMICAL CONVERSION      62
-----
  PRECURSOR:      3   1   2   1   5   1
  *BYPRODUCT:     3   4   2   1   5   1   YIELD =  0.11416   RATE =  0.01461
  BYPRODUCT:      5   4   2   1   3   1   YIELD =  0.04496
  BYPRODUCT:      3   1   2   1   5   4   YIELD =  0.84089

-----
--- CHEMICAL CONVERSION      63
-----
  PRECURSOR:      3   4   2   1   2   1
  BYPRODUCT:      3   1   2   1   5   4   YIELD =  0.50000
  *BYPRODUCT:     3   4   2   1   5   1   YIELD =  0.50000   RATE =  1.00000

-----
--- CHEMICAL CONVERSION      64
-----
  PRECURSOR:      3   1   5   1   1   1
  *BYPRODUCT:     3   1   5   1   1   4   YIELD =  0.32320   RATE =  0.48701
  BYPRODUCT:      3   1   5   1   4   1   YIELD =  0.00010
  BYPRODUCT:      3   1   5   4   1   1   YIELD =  0.13729
  BYPRODUCT:      3   4   5   1   1   1   YIELD =  0.54941

-----
--- CHEMICAL CONVERSION      65
-----
  PRECURSOR:      3   4   1   1   2   1
  *BYPRODUCT:     3   1   5   1   1   4   YIELD =  1.00000   RATE =  1.00000

-----
--- CHEMICAL CONVERSION      66
-----
  PRECURSOR:      3   1   4   1   1   1
  *BYPRODUCT:     3   1   5   1   1   1   YIELD =  1.00000   RATE =  1.00000

-----
--- CHEMICAL CONVERSION      67
-----
  PRECURSOR:      5   4   1   1   2   1
  *BYPRODUCT:     3   1   2   1   1   4   YIELD =  0.85714   RATE =  1.00000
  BYPRODUCT:      9   4   1   1   2   1   YIELD =  0.14286

-----
--- CHEMICAL CONVERSION      68
-----

```

PRECURSOR:	3	1	2	1	1	1			
*BYPRODUCT:	3	1	2	1	1	4	YIELD =	0.76069	RATE = 0.00001
BYPRODUCT:	3	1	2	1	4	1	YIELD =	0.02309	
BYPRODUCT:	3	1	2	4	1	1	YIELD =	0.04067	
BYPRODUCT:	3	4	2	1	1	1	YIELD =	0.17554	

--- CHEMICAL CONVERSION	69								

PRECURSOR:	3	4	5	1	2	1			
*BYPRODUCT:	3	1	2	1	1	4	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION	70								

PRECURSOR:	3	4	1	5	2	1			
*BYPRODUCT:	3	1	2	1	1	4	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION	71								

PRECURSOR:	3	4	1	1	2	5			
*BYPRODUCT:	3	1	2	1	1	4	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION	72								

PRECURSOR:	2	1	5	1	1	1			
*BYPRODUCT:	2	1	5	4	1	1	YIELD =	0.73178	RATE =
1159.73000	BYPRODUCT:	2	1	5	1	4	1	YIELD =	0.00000
BYPRODUCT:	2	1	5	1	1	4	YIELD =	0.09934	
BYPRODUCT:	2	4	5	1	1	1	YIELD =	0.16887	

--- CHEMICAL CONVERSION	73								

PRECURSOR:	2	4	1	1	2	1			
*BYPRODUCT:	2	1	5	4	1	1	YIELD =	0.50000	RATE = 1.00000
BYPRODUCT:	2	1	5	1	1	4	YIELD =	0.50000	

--- CHEMICAL CONVERSION	74								

PRECURSOR:	2	1	5	1	1	1			
*BYPRODUCT:	3	1	2	1	1	1	YIELD =	0.85714	RATE = 1.00000
BYPRODUCT:	9	1	2	1	1	1	YIELD =	0.14286	

--- CHEMICAL CONVERSION	75								

PRECURSOR:	3	5	2	1	1	1			
*BYPRODUCT:	3	1	2	1	1	1	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION	76								

PRECURSOR:	3	1	2	5	1	1			
*BYPRODUCT:	3	1	2	1	1	1	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION	77								

PRECURSOR:	3	1	2	1	5	1			
*BYPRODUCT:	3	1	2	1	1	1	YIELD =	1.00000	RATE = 1.00000

```

-----
--- CHEMICAL CONVERSION      78
-----
  PRECURSOR:      3   1   2   1   1   5
  *BYPRODUCT:     3   1   2   1   1   1   YIELD =  1.00000   RATE = 1.00000

```

```

-----
--- CHEMICAL CONVERSION      79
-----
  PRECURSOR:      3   1   2   1   5   1
  *BYPRODUCT:     3   1   2   1   5   4   YIELD =  0.84089   RATE = 0.10762
  BYPRODUCT:      5   4   2   1   3   1   YIELD =  0.04496
  BYPRODUCT:      3   4   2   1   5   1   YIELD =  0.11416

```

```

-----
--- CHEMICAL CONVERSION      80
-----
  PRECURSOR:      3   4   2   1   2   1
  *BYPRODUCT:     3   1   2   1   5   4   YIELD =  0.50000   RATE = 1.00000
  BYPRODUCT:      3   4   2   1   5   1   YIELD =  0.50000

```

A6.3.4. Run #4: 2-Amino-6-Chlorobenzonitrile

```

-----
--- CHEMICAL CONVERSION      1
-----
  PRECURSOR:      3   4   1   1   1   2
  *BYPRODUCT:     3   5   1   1   1   2   YIELD =  1.00000   RATE = 1.00000

```

```

-----
--- CHEMICAL CONVERSION      2
-----
  PRECURSOR:      5   4   1   1   1   2
  *BYPRODUCT:     3   4   1   1   1   2   YIELD =  0.85714   RATE = 1.00000
  BYPRODUCT:      9   4   1   1   1   2   YIELD =  0.14286

```

```

-----
--- CHEMICAL CONVERSION      3
-----
  PRECURSOR:      3   4   5   1   1   2
  *BYPRODUCT:     3   4   1   1   1   2   YIELD =  1.00000   RATE = 1.00000

```

```

-----
--- CHEMICAL CONVERSION      4
-----
  PRECURSOR:      3   4   1   5   1   2
  *BYPRODUCT:     3   4   1   1   1   2   YIELD =  1.00000   RATE = 1.00000

```

```

-----
--- CHEMICAL CONVERSION      5
-----
  PRECURSOR:      3   4   1   1   5   2
  *BYPRODUCT:     3   4   1   1   1   2   YIELD =  1.00000   RATE = 1.00000

```

```

-----
--- CHEMICAL CONVERSION      6
-----
  PRECURSOR:      5   1   1   1   1   2
  *BYPRODUCT:     5   4   1   1   1   2   YIELD =  0.62683   RATE = 7.49364
  BYPRODUCT:      5   1   4   1   1   2   YIELD =  0.00359
  BYPRODUCT:      5   1   1   4   1   2   YIELD =  0.36875
  BYPRODUCT:      5   1   1   1   4   2   YIELD =  0.00083

```

```

-----
--- CHEMICAL CONVERSION      7

```

PRECURSOR:	2	4	1	1	1	2				
*BYPRODUCT:	5	4	1	1	1	2	YIELD =	1.00000	RATE = 1.00000	
BYPRODUCT:	5	1	1	1	4	2	YIELD =	0.00000		

---	CHEMICAL CONVERSION	8								

PRECURSOR:	5	4	1	1	1	1				
BYPRODUCT:	4	5	1	1	1	2	YIELD =	0.00000		
BYPRODUCT:	5	4	1	2	1	1	YIELD =	0.80386		
BYPRODUCT:	5	4	1	1	2	1	YIELD =	0.00000		
*BYPRODUCT:	5	4	1	1	1	2	YIELD =	0.19614	RATE = 0.01848	
BYPRODUCT:	5	4	1	2	1	2	YIELD =	0.00000		

---	CHEMICAL CONVERSION	9								

PRECURSOR:	4	1	1	1	1	2				
*BYPRODUCT:	5	1	1	1	1	2	YIELD =	1.00000	RATE = 1.00000	

---	CHEMICAL CONVERSION	10								

PRECURSOR:	5	1	1	1	1	1				
*BYPRODUCT:	5	1	1	1	1	2	YIELD =	0.32796	RATE =	
1220000.00000	BYPRODUCT:	5	1	2	1	1	YIELD =	0.00000		
BYPRODUCT:	5	1	1	2	1	1	YIELD =	0.67204		

---	CHEMICAL CONVERSION	11								

PRECURSOR:	1	1	1	1	1	2				
*BYPRODUCT:	4	1	1	1	1	2	YIELD =	0.31302	RATE = 0.06000	
BYPRODUCT:	1	4	1	1	1	2	YIELD =	0.00876		
BYPRODUCT:	1	1	4	1	1	2	YIELD =	0.67821		

---	CHEMICAL CONVERSION	12								

PRECURSOR:	4	5	1	1	1	2				
*BYPRODUCT:	4	1	1	1	1	2	YIELD =	1.00000	RATE = 1.00000	

---	CHEMICAL CONVERSION	13								

PRECURSOR:	4	1	5	1	1	2				
*BYPRODUCT:	4	1	1	1	1	2	YIELD =	1.00000	RATE = 1.00000	

---	CHEMICAL CONVERSION	14								

PRECURSOR:	4	1	1	5	1	2				
*BYPRODUCT:	4	1	1	1	1	2	YIELD =	1.00000	RATE = 1.00000	

---	CHEMICAL CONVERSION	15								

PRECURSOR:	4	1	1	1	5	2				
*BYPRODUCT:	4	1	1	1	1	2	YIELD =	1.00000	RATE = 1.00000	

---	CHEMICAL CONVERSION	16								

PRECURSOR:	4	1	1	1	1	1				
*BYPRODUCT:	5	1	1	1	1	1	YIELD =	1.00000	RATE = 1.00000	

```

-----
--- CHEMICAL CONVERSION      17
-----
PRECURSOR:      2   1   1   1   1   2
*BYPRODUCT:    2   4   1   1   1   2   YIELD =  0.18750   RATE = 0.00005
BYPRODUCT:     2   1   4   1   1   2   YIELD =  0.81250

-----
--- CHEMICAL CONVERSION      18
-----
PRECURSOR:      1   1   1   1   1   2
*BYPRODUCT:    2   1   1   1   1   2   YIELD =  0.36291   RATE = 0.03800
BYPRODUCT:     1   2   1   1   1   2   YIELD =  0.01249
BYPRODUCT:     1   1   2   1   1   2   YIELD =  0.62459

-----
--- CHEMICAL CONVERSION      19
-----
PRECURSOR:      2   5   1   1   1   2
*BYPRODUCT:    2   1   1   1   1   2   YIELD =  1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      20
-----
PRECURSOR:      2   1   5   1   1   2
*BYPRODUCT:    2   1   1   1   1   2   YIELD =  1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      21
-----
PRECURSOR:      5   1   1   1   1   1
*BYPRODUCT:    5   4   1   1   1   1   YIELD =  0.77269   RATE =
17842.00000
BYPRODUCT:     5   1   4   1   1   1   YIELD =  0.00003
BYPRODUCT:     5   1   1   4   1   1   YIELD =  0.22728

-----
--- CHEMICAL CONVERSION      22
-----
PRECURSOR:      4   1   1   1   1   2
*BYPRODUCT:    5   4   1   1   1   1   YIELD =  1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      23
-----
PRECURSOR:      1   1   1   1   1   2
*BYPRODUCT:    4   1   1   1   1   2   YIELD =  0.31302   RATE = 0.06000
BYPRODUCT:     1   4   1   1   1   2   YIELD =  0.00876
BYPRODUCT:     1   1   4   1   1   2   YIELD =  0.67821

-----
--- CHEMICAL CONVERSION      24
-----
PRECURSOR:      4   5   1   1   1   2
*BYPRODUCT:    4   1   1   1   1   2   YIELD =  1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      25
-----
PRECURSOR:      4   1   5   1   1   2
*BYPRODUCT:    4   1   1   1   1   2   YIELD =  1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      26
-----

```

PRECURSOR:	4	1	1	5	1	2			
*BYPRODUCT:	4	1	1	1	1	2	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION	27								

PRECURSOR:	4	1	1	1	5	2			
*BYPRODUCT:	4	1	1	1	1	2	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION	28								

PRECURSOR:	3	1	5	1	1	2			
*BYPRODUCT:	3	4	5	1	1	2	YIELD =	0.80783	RATE = 0.00070
BYPRODUCT:	3	1	5	4	1	2	YIELD =	0.18716	
BYPRODUCT:	3	1	5	1	4	2	YIELD =	0.00501	

--- CHEMICAL CONVERSION	29								

PRECURSOR:	3	4	2	1	1	2			
*BYPRODUCT:	3	4	5	1	1	2	YIELD =	1.00000	RATE = 1.00000
BYPRODUCT:	3	4	2	1	1	5	YIELD =	0.00000	

--- CHEMICAL CONVERSION	30								

PRECURSOR:	3	1	4	1	1	2			
*BYPRODUCT:	3	1	5	1	1	2	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION	31								

PRECURSOR:	3	1	5	1	1	1			
BYPRODUCT:	3	1	1	1	5	2	YIELD =	0.18694	
BYPRODUCT:	5	1	3	1	1	2	YIELD =	0.04691	
BYPRODUCT:	3	1	5	1	2	1	YIELD =	0.00000	
*BYPRODUCT:	3	1	5	1	1	2	YIELD =	0.76615	RATE = 1.34500

--- CHEMICAL CONVERSION	32								

PRECURSOR:	5	1	4	1	1	2			
*BYPRODUCT:	3	1	4	1	1	2	YIELD =	0.85714	RATE = 1.00000
BYPRODUCT:	9	1	4	1	1	2	YIELD =	0.14286	

--- CHEMICAL CONVERSION	33								

PRECURSOR:	3	1	1	1	1	2			
BYPRODUCT:	3	4	1	1	1	2	YIELD =	0.00112	
*BYPRODUCT:	3	1	4	1	1	2	YIELD =	0.81138	RATE = 0.00006
BYPRODUCT:	3	1	1	4	1	2	YIELD =	0.00026	
BYPRODUCT:	3	1	1	1	4	2	YIELD =	0.18724	

--- CHEMICAL CONVERSION	34								

PRECURSOR:	3	5	4	1	1	2			
*BYPRODUCT:	3	1	4	1	1	2	YIELD =	1.00000	RATE = 1.00000

--- CHEMICAL CONVERSION	35								

PRECURSOR:	3	1	4	5	1	2			
*BYPRODUCT:	3	1	4	1	1	2	YIELD =	1.00000	RATE = 1.00000

```

-----
--- CHEMICAL CONVERSION      36
-----
  PRECURSOR:      3   1   4   1   5   2
  *BYPRODUCT:     3   1   4   1   1   2   YIELD =  1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      37
-----
  PRECURSOR:      3   1   4   1   1   1
  *BYPRODUCT:     3   1   5   1   1   1   YIELD =  1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      38
-----
  PRECURSOR:      3   1   1   1   5   2
  *BYPRODUCT:     3   4   1   1   5   2   YIELD =  0.69466   RATE = 0.00041
  BYPRODUCT:      3   1   4   1   5   2   YIELD =  0.03176
  BYPRODUCT:      5   4   1   1   3   2   YIELD =  0.27358

-----
--- CHEMICAL CONVERSION      39
-----
  PRECURSOR:      3   4   1   1   2   2
  *BYPRODUCT:     3   4   1   1   5   2   YIELD =  1.00000   RATE = 1.00000
  BYPRODUCT:      3   4   1   1   2   5   YIELD =  0.00000

-----
--- CHEMICAL CONVERSION      40
-----
  PRECURSOR:      3   1   1   1   4   2
  *BYPRODUCT:     3   1   1   1   5   2   YIELD =  1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      41
-----
  PRECURSOR:      3   1   5   1   1   1
  BYPRODUCT:      3   1   5   1   1   2   YIELD =  0.76615
  BYPRODUCT:      3   1   5   1   2   1   YIELD =  0.00000
  BYPRODUCT:      5   1   3   1   1   2   YIELD =  0.04691
  *BYPRODUCT:     3   1   1   1   5   2   YIELD =  0.18694   RATE = 0.32818

-----
--- CHEMICAL CONVERSION      42
-----
  PRECURSOR:      4   1   1   1   5   2
  *BYPRODUCT:     3   1   1   1   4   2   YIELD =  0.85714   RATE = 1.00000
  BYPRODUCT:      9   1   1   1   4   2   YIELD =  0.14286

-----
--- CHEMICAL CONVERSION      43
-----
  PRECURSOR:      3   1   1   1   1   2
  BYPRODUCT:      3   4   1   1   1   2   YIELD =  0.00112
  BYPRODUCT:      3   1   4   1   1   2   YIELD =  0.81138
  BYPRODUCT:      3   1   1   4   1   2   YIELD =  0.00026
  *BYPRODUCT:     3   1   1   1   4   2   YIELD =  0.18724   RATE = 0.00001

-----
--- CHEMICAL CONVERSION      44
-----
  PRECURSOR:      3   5   1   1   4   2
  *BYPRODUCT:     3   1   1   1   4   2   YIELD =  1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      45

```



```

-----
  PRECURSOR:      3  1  5  1  4  2
  *BYPRODUCT:    3  1  1  1  4  2   YIELD =  1.00000   RATE = 1.00000

```

```

-----
--- CHEMICAL CONVERSION      46
-----

```

```

  PRECURSOR:      3  1  1  5  4  2
  *BYPRODUCT:    3  1  1  1  4  2   YIELD =  1.00000   RATE = 1.00000

```

A6.3.5. Run #5: 2-Chloro-4-Aminobenzonitrile

```

-----
--- CHEMICAL CONVERSION      1
-----

```

```

  PRECURSOR:      3  2  1  4  1  1
  *BYPRODUCT:    3  2  1  5  1  1   YIELD =  1.00000   RATE = 1.00000

```

```

-----
--- CHEMICAL CONVERSION      2
-----

```

```

  PRECURSOR:      5  2  1  4  1  1
  *BYPRODUCT:    3  2  1  4  1  1   YIELD =  0.85714   RATE = 1.00000
  BYPRODUCT:     9  2  1  4  1  1   YIELD =  0.14286

```

```

-----
--- CHEMICAL CONVERSION      3
-----

```

```

  PRECURSOR:      3  2  5  4  1  1
  *BYPRODUCT:    3  2  1  4  1  1   YIELD =  1.00000   RATE = 1.00000

```

```

-----
--- CHEMICAL CONVERSION      4
-----

```

```

  PRECURSOR:      3  2  1  4  5  1
  *BYPRODUCT:    3  2  1  4  1  1   YIELD =  1.00000   RATE = 1.00000

```

```

-----
--- CHEMICAL CONVERSION      5
-----

```

```

  PRECURSOR:      3  2  1  4  1  5
  *BYPRODUCT:    3  2  1  4  1  1   YIELD =  1.00000   RATE = 1.00000

```

```

-----
--- CHEMICAL CONVERSION      6
-----

```

```

  PRECURSOR:      5  2  1  1  1  1
  BYPRODUCT:     5  2  4  1  1  1   YIELD =  0.00083
  *BYPRODUCT:    5  2  1  4  1  1   YIELD =  0.36875   RATE = 4.40832
  BYPRODUCT:     5  2  1  1  4  1   YIELD =  0.00359
  BYPRODUCT:     5  2  1  1  1  4   YIELD =  0.62683

```

```

-----
--- CHEMICAL CONVERSION      7
-----

```

```

  PRECURSOR:      2  2  1  4  1  1
  *BYPRODUCT:    5  2  1  4  1  1   YIELD =  1.00000   RATE = 1.00000
  BYPRODUCT:     5  2  1  1  4  1   YIELD =  0.00000

```

```

-----
--- CHEMICAL CONVERSION      8
-----

```

```

  PRECURSOR:      5  1  1  4  1  1
  *BYPRODUCT:    5  2  1  4  1  1   YIELD =  1.00000   RATE = 0.03697
  BYPRODUCT:     4  2  1  5  1  1   YIELD =  0.00000
  BYPRODUCT:     5  2  1  4  1  2   YIELD =  0.00000

```

```

-----
--- CHEMICAL CONVERSION          9
-----
  PRECURSOR:      4  2  1  1  1  1
  *BYPRODUCT:     5  2  1  1  1  1   YIELD =  1.00000   RATE =  1.00000

-----
--- CHEMICAL CONVERSION          10
-----
  PRECURSOR:      5  1  1  1  1  1
  *BYPRODUCT:     5  2  1  1  1  1   YIELD =  0.32796   RATE =
1220000.00000
  BYPRODUCT:     5  1  2  1  1  1   YIELD =  0.00000
  BYPRODUCT:     5  1  1  2  1  1   YIELD =  0.67204

-----
--- CHEMICAL CONVERSION          11
-----
  PRECURSOR:      1  2  1  1  1  1
  *BYPRODUCT:     4  2  1  1  1  1   YIELD =  0.31302   RATE =  0.06000
  BYPRODUCT:     1  2  1  4  1  1   YIELD =  0.00876
  BYPRODUCT:     1  2  1  1  4  1   YIELD =  0.67821

-----
--- CHEMICAL CONVERSION          12
-----
  PRECURSOR:      4  2  5  1  1  1
  *BYPRODUCT:     4  2  1  1  1  1   YIELD =  1.00000   RATE =  1.00000

-----
--- CHEMICAL CONVERSION          13
-----
  PRECURSOR:      4  2  1  5  1  1
  *BYPRODUCT:     4  2  1  1  1  1   YIELD =  1.00000   RATE =  1.00000

-----
--- CHEMICAL CONVERSION          14
-----
  PRECURSOR:      4  2  1  1  5  1
  *BYPRODUCT:     4  2  1  1  1  1   YIELD =  1.00000   RATE =  1.00000

-----
--- CHEMICAL CONVERSION          15
-----
  PRECURSOR:      4  2  1  1  1  5
  *BYPRODUCT:     4  2  1  1  1  1   YIELD =  1.00000   RATE =  1.00000

-----
--- CHEMICAL CONVERSION          16
-----
  PRECURSOR:      4  1  1  1  1  1
  *BYPRODUCT:     5  1  1  1  1  1   YIELD =  1.00000   RATE =  1.00000

-----
--- CHEMICAL CONVERSION          17
-----
  PRECURSOR:      2  2  1  1  1  1
  BYPRODUCT:      2  2  4  1  1  1   YIELD =  0.18750
  *BYPRODUCT:     2  2  1  4  1  1   YIELD =  0.81250   RATE =  0.00022

-----
--- CHEMICAL CONVERSION          18
-----
  PRECURSOR:      1  2  1  1  1  1
  *BYPRODUCT:     2  2  1  1  1  1   YIELD =  0.36291   RATE =  0.03800
  BYPRODUCT:      1  2  1  2  1  1   YIELD =  0.01249
  BYPRODUCT:      1  2  1  1  2  1   YIELD =  0.62459

```

```

-----
--- CHEMICAL CONVERSION      19
-----
  PRECURSOR:      2   2   5   1   1   1
  *BYPRODUCT:    2   2   1   1   1   1   YIELD = 1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      20
-----
  PRECURSOR:      2   2   1   5   1   1
  *BYPRODUCT:    2   2   1   1   1   1   YIELD = 1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      21
-----
  PRECURSOR:      5   1   1   1   1   1
  BYPRODUCT:     5   4   1   1   1   1   YIELD = 0.77269
  BYPRODUCT:     5   1   4   1   1   1   YIELD = 0.00003
  *BYPRODUCT:    5   1   1   4   1   1   YIELD = 0.22728   RATE =
5248.00000

-----
--- CHEMICAL CONVERSION      22
-----
  PRECURSOR:      2   1   1   4   1   1
  *BYPRODUCT:    5   1   1   4   1   1   YIELD = 1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      23
-----
  PRECURSOR:      1   2   1   1   1   1
  BYPRODUCT:     4   2   1   1   1   1   YIELD = 0.31302
  BYPRODUCT:     1   2   1   4   1   1   YIELD = 0.00876
  *BYPRODUCT:    1   2   1   1   4   1   YIELD = 0.67821   RATE = 0.13000

-----
--- CHEMICAL CONVERSION      24
-----
  PRECURSOR:      2   5   1   4   1   1
  *BYPRODUCT:    1   2   1   1   4   1   YIELD = 1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      25
-----
  PRECURSOR:      2   1   5   4   1   1
  *BYPRODUCT:    1   2   1   1   4   1   YIELD = 1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      26
-----
  PRECURSOR:      3   2   5   1   1   1
  *BYPRODUCT:    3   2   5   4   1   1   YIELD = 0.27358   RATE = 0.00016
  BYPRODUCT:     3   2   5   1   4   1   YIELD = 0.03176
  BYPRODUCT:     5   2   3   4   1   1   YIELD = 0.69466

-----
--- CHEMICAL CONVERSION      27
-----
  PRECURSOR:      3   2   2   4   1   1
  BYPRODUCT:     3   5   2   4   1   1   YIELD = 0.00000
  *BYPRODUCT:    3   2   5   4   1   1   YIELD = 1.00000   RATE = 1.00000

-----
--- CHEMICAL CONVERSION      28
-----

```

PRECURSOR:	3	2	4	1	1	1				
*BYPRODUCT:	3	2	5	1	1	1	YIELD =	1.00000	RATE = 1.00000	

---	CHEMICAL CONVERSION	29								

PRECURSOR:	3	1	5	1	1	1				
*BYPRODUCT:	3	2	5	1	1	1	YIELD =	0.18694	RATE = 0.32818	
BYPRODUCT:	3	1	5	2	1	1	YIELD =	0.04691		
BYPRODUCT:	3	1	5	1	2	1	YIELD =	0.00000		
BYPRODUCT:	3	2	1	1	5	1	YIELD =	0.76615		

---	CHEMICAL CONVERSION	30								

PRECURSOR:	4	2	5	1	1	1				
*BYPRODUCT:	3	2	4	1	1	1	YIELD =	0.85714	RATE = 1.00000	
BYPRODUCT:	9	2	4	1	1	1	YIELD =	0.14286		

---	CHEMICAL CONVERSION	31								

PRECURSOR:	3	2	1	1	1	1				
*BYPRODUCT:	3	2	4	1	1	1	YIELD =	0.18724	RATE = 0.00001	
BYPRODUCT:	3	2	1	4	1	1	YIELD =	0.00026		
BYPRODUCT:	3	2	1	1	4	1	YIELD =	0.81138		
BYPRODUCT:	3	2	1	1	1	4	YIELD =	0.00112		

---	CHEMICAL CONVERSION	32								

PRECURSOR:	3	2	4	5	1	1				
*BYPRODUCT:	3	2	4	1	1	1	YIELD =	1.00000	RATE = 1.00000	

---	CHEMICAL CONVERSION	33								

PRECURSOR:	3	2	4	1	5	1				
*BYPRODUCT:	3	2	4	1	1	1	YIELD =	1.00000	RATE = 1.00000	

---	CHEMICAL CONVERSION	34								

PRECURSOR:	3	2	4	1	1	5				
*BYPRODUCT:	3	2	4	1	1	1	YIELD =	1.00000	RATE = 1.00000	

---	CHEMICAL CONVERSION	35								

PRECURSOR:	3	1	4	1	1	1				
*BYPRODUCT:	3	1	5	1	1	1	YIELD =	1.00000	RATE = 1.00000	

---	CHEMICAL CONVERSION	36								

PRECURSOR:	3	2	1	1	5	1				
BYPRODUCT:	3	2	4	1	5	1	YIELD =	0.00501		
*BYPRODUCT:	3	1	5	4	1	2	YIELD =	0.18716	RATE = 0.00016	
BYPRODUCT:	3	2	1	1	5	4	YIELD =	0.80783		

---	CHEMICAL CONVERSION	37								

PRECURSOR:	3	2	1	4	2	1				
BYPRODUCT:	3	5	1	4	2	1	YIELD =	0.00000		
*BYPRODUCT:	3	1	5	4	1	2	YIELD =	1.00000	RATE = 1.00000	

```

-----
--- CHEMICAL CONVERSION      38
-----
  PRECURSOR:      3  2  1  1  4  1
  *BYPRODUCT:    3  2  1  1  5  1   YIELD =  1.00000   RATE = 1.00000

```

```

-----
--- CHEMICAL CONVERSION      39
-----
  PRECURSOR:      3  1  5  1  1  1
  *BYPRODUCT:    3  2  1  1  5  1   YIELD =  0.76615   RATE = 1.34500
  BYPRODUCT:     3  1  5  1  2  1   YIELD =  0.00000
  BYPRODUCT:     3  1  5  2  1  1   YIELD =  0.04691
  BYPRODUCT:     3  2  5  1  1  1   YIELD =  0.18694

```

```

-----
--- CHEMICAL CONVERSION      40
-----
  PRECURSOR:      2  5  1  4  1  1
  *BYPRODUCT:    3  2  1  1  4  1   YIELD =  0.85714   RATE = 1.00000
  BYPRODUCT:     9  2  1  1  4  1   YIELD =  0.14286

```

```

-----
--- CHEMICAL CONVERSION      41
-----
  PRECURSOR:      3  2  1  1  1  1
  BYPRODUCT:     3  2  4  1  1  1   YIELD =  0.18724
  BYPRODUCT:     3  2  1  4  1  1   YIELD =  0.00026
  *BYPRODUCT:    3  2  1  1  4  1   YIELD =  0.81138   RATE = 0.00006
  BYPRODUCT:     3  2  1  1  1  4   YIELD =  0.00112

```

```

-----
--- CHEMICAL CONVERSION      42
-----
  PRECURSOR:      3  2  5  1  4  1
  *BYPRODUCT:    3  2  1  1  4  1   YIELD =  1.00000   RATE = 1.00000

```

```

-----
--- CHEMICAL CONVERSION      43
-----
  PRECURSOR:      3  2  1  5  4  1
  *BYPRODUCT:    3  2  1  1  4  1   YIELD =  1.00000   RATE = 1.00000

```

```

-----
--- CHEMICAL CONVERSION      44
-----
  PRECURSOR:      3  2  1  1  4  5
  *BYPRODUCT:    3  2  1  1  4  1   YIELD =  1.00000   RATE = 1.00000

```

References

- Abraham, M. and P. Grellier, "Substitution at Saturated Carbon. Part XX," *J. Chem. Soc. Perkin II* (1976), 1735.
- Abrams, D. S. and J. M. Prausnitz, "Statistical Thermodynamics of Liquid Mixtures: A New Expression for the Excess Gibbs Energy of Partly or Completely Miscible Systems," *AIChE J.*, **21**, 116 (1975).
- Agnihotri, R. and R. Motard, "Reaction Path Synthesis in Industrial Chemistry," *Computer Applications to Chemical Process Design and Simulation, ACS Symposium Series*, **124**, 193 (1980).
- Andreovich, M. and A. Westerberg, "A Simple Synthesis Method Based on Utility Bounding for Heat-Integrated Distillation Sequences," *AIChE J.*, **31**, 363 (1985).
- Aumond, J. *Ph.D. Proposal*, Massachusetts Institute of Technology (1994).
- Austin, G., *Shreve's Chemical Process Industries*, Fifth Ed., McGraw-Hill, New York (1984).
- Barton, P. and C. Pantelides, "Equation-Oriented Dynamic Simulation—Current Status and Future Perspectives," *Comp. Chem. Eng., Supplement*, **17**, S263 (1993).
- Baudet, H., "The Replaceability of the Halogen Atom in 1-Chloro- and 1-Bromo-2-Cyano-4-Nitrobenzene," *J. Rec. Trav. Chim.*, **43**, 708 (1924).
- Baum, R., "Combinatorial Approaches Provide Fresh Leads for Medicinal Chemistry," *Chem. and Eng. News*, 20 (February 7, 1994).
- Baumer, L., G. Sala, and G. Sello, "Organic Synthesis Planning: An Algorithm for Selection of Strategic Bond Forming Sequences," *Tetrahedron*, **45**, 2665 (1989).
- Baumer, L., G. Sala, and G. Sello, "The LILITH Approach to Organic Synthesis Planning," *Anal. Chim. Acta*, **235**, 209 (1990).
- Berglund, R. and C. Lawson, "Preventing Pollution in the CPI," *Chem. Eng.*, 120, (September, 1991).
- Bersohn, M and A. Esack, "Computers and Organic Synthesis," *Chem. Rev.*, **76**, 269 (1976).
- Birewar, D. and I. Grossmann, "Efficient Optimization Algorithms for Zero-Wait Scheduling of Multiproduct Batch Plants," *Ind. Eng. Chem. Res.*, **28**, 1333 (1989).
- Booth, G., "Nitro Compounds, Aromatic," *Ullmann's Encyclopedia of Industrial Chemistry, Fifth Ed., Vol A17*, pp 411-455, B. Elvers, S. Hawkins, G. Schuly, Eds., VCH, Weinheim (1985).
- Brinck, T., J. Murray, and P. Politzer, "Octanol/Water Partition Coefficients Expressed in Terms of Solute Molecular Surface Areas and Electrostatic Potentials," *J. Org. Chem.*, **58**, 7070 (1993).
- Brooke, A., D. Kendrick, and A. Meeraus, *GAMS: A User's Guide*, Scientific Press, Palo Alto (1988).
- Buehler, C. and D. Pearson, *Survey of Organic Syntheses. Vol. II*, John Wiley and Sons, New York (1977).
- Bullard, L. and L. Biegler, "Iterative Linear Programming Strategies for Constrained Simulation," *Comp. Chem. Eng.*, **15**, 239 (1991).
- Carbeck, J. and G. Rutledge, "A Simple Approach to Electronic Charge Induction in Atomistic Simulations. I. Model Development and Parameterization," *J. Computer-Aided Materials Design*, **1**, 97 (1993).
- Chung, Y. and A. Westerberg, "A Proposed Numerical Algorithm for Solving Nonlinear Index Problems," *Ind. Eng. Chem. Res.*, **29**, 1234 (1990).

- Ciric, A. and T. Jia, "Economic Sensitivity Analysis of Waste Treatment Costs in Source Reduction Projects: Continuous Optimization Problems," *Comp. Chem. Eng.*, **18**, 481 (1994).
- Clay, R. *Ph.D. Thesis*, Carnegie Mellon University (1994).
- Corey, E., "Centenary Lecture: Computer-Assisted Analysis of Complex Synthetic Problems," *Chem. Soc. Rev.*, 455 (1971).
- Corey, E. and X. Cheng, *The Logic of Chemical Synthesis*, John Wiley and Sons, New York (1989).
- Corey, E., R. Cramer, and W. Howe, "Computer-Assisted Synthetic Analysis for Complex Molecules. Methods and Procedures for Machine Generation of Synthetic Intermediates," *J. Am. Chem. Soc.*, **94**, 440 (1972).
- Corey, E., W. Howe, H. Orf *et al.*, "General Methods of Synthetic Analysis. Strategic Bond Disconnections for Bridged Polycyclic Structures," *J. Am. Chem. Soc.*, **97**, 6116 (1975).
- Corey, E., W. Howe, and D. Pensak, "Computer-Assisted Synthetic Analysis. Methods for Machine Generation of Synthetic Intermediates Involving Multistep Look-Ahead," *J. Am. Chem. Soc.*, **96**, 7724 (1974).
- Corey, E., A. Johnson, and A. Long, "Computer-Assisted Synthetic Analysis. Techniques for Efficient Long-Range Retrosynthetic Searches Applied to Robinson Annulation Processes," *J. Org. Chem.*, **45**, 2051 (1980).
- Corey, E. and W. Jorgensen, "Computer-Assisted Synthetic Analysis. Synthetic Strategies Based on Appendiges and the Use of Reconnective Transforms," *J. Am. Chem. Soc.*, **98**, 189 (1976).
- Corey, E. and A. Long, "Computer-Assisted Synthetic Analysis. Techniques for Efficient Long-Range Retrosynthetic Search Strategies for Stereoselective Olefin Synthesis," *J. Org. Chem.*, **43**, 2208 (1978).
- Corey, E. and W. Wipke, "Computer-Assisted Design of Complex Organic Synthesis," *Science*, **166**, 178 (10 October 1969).
- Cox, P. and A. Strachan, "Two Phase Nitration of Chlorobenzene," *Chem. Engng. Sci.*, **26**, 1013 (1971).
- Cox, P. and A. Strachan, "Two Phase Nitration of Toluene—I," *Chem. Engng. Sci.*, **27**, 457 (1972).
- Cox, P. and A. Strachan, "Two Phase Nitration of Toluene. Part II," *Chem. Engng. J.*, **4**, 253 (1972).
- Cuthrell, J. and L. Biegler, "On the Optimization of Differential-Algebraic Process Systems," *AIChE J.*, **33**, 1257 (1987).
- De La Mare, P. and J. Ridd, *Aromatic Substitution: Nitration and Halogenation*, Butterworths Scientific Publishers, London (1959).
- Dewar, M. and W. Thiel, "Ground States of Molecules. 39. MNDO Results for Molecules Containing Hydrogen, Carbon, Nitrogen, and Oxygen," *J. Am. Chem. Soc.*, **99**, 4899 (1977).
- Dewar, M., E. Zuebisch, E. Healy, *et al.*, "AM1: A New General Purpose Quantum Mechanical Molecular Model," *J. Am. Chem. Soc.*, **107**, 3902 (1985).
- Dey, B. and Y. Doraiswami, "The Reactivity of the Chlorine Atom in the Benzene Nucleus," *J. Indian Chem. Soc.*, **10**, 309 (1933).
- Dixon, D., "Numerical Simulation of Molecular Properties in the Chemical Industry," pp 169-194, *Science and Engineering on Cray Supercomputers*, Cray Research, Minneapolis (1987).
- Diwekar, U. and K. Madhavan, "Multicomponent Batch Distillation Column Design," *Ind. Eng. Chem. Res.*, **30**, 713 (1991).
- Doucette, W., "Measurement and Estimation of Octanol/Water Partition Coefficients and Aqueous Solubilities for Halogenated Hydrocarbons," *Ph.D. Thesis*, University of Wisconsin-Madison (1985).

- Douglas, J., *Conceptual Design of Chemical Processes*, McGraw-Hill, New York (1988).
- Douglas, J., "Synthesis of Multistep Reaction Processes," *Foundations of Computer-Aided Process Design*, pp 79-103, J. Siirola, I. Grossmann, G. Stephanopoulos, Eds., Elsevier Publishing Company, New York (1989).
- Douglas, J., "Process Synthesis for Waste Minimization," *Ind. Eng. Chem. Res.*, **31**, 238 (1992).
- Duran, M. and I. Grossmann, "An Outer-Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programs," *Math. Prog.*, **36**, 307 (1986).
- Eastman Chemical Co., "Your Partner In Innovation: Eastman Fine Chemicals," *Chem. and Eng. News*, 56 (October 18, 1994).
- Eastman Chemical Co., "Your Partner In Innovation: Fine Chemicals From Eastman," *Chem. and Eng. News*, 26 (April 11, 1994).
- Eckenfelder, W., *Industrial Water Pollution Control, Second Ed.*, McGraw-Hill, New York (1989).
- Eisenhauer, J. and S. McQueen, *Environmental Considerations in Process Design and Simulation*, Environmental Protection Agency, Department of Energy, Center for Waste Reduction Technologies, et al., Cincinnati, OH (1992).
- El-Halwagi, M., "Synthesis of Reverse-Osmosis Networks for Waste Reduction," *AIChE J.*, **38**, 1185 (1992).
- El-Halwagi, M. and V. Manousiouthakis, "Simultaneous Synthesis of Mass Exchange and Regeneration Networks," *AIChE J.*, **36**, 1209 (1990).
- El-Halwagi, M. and B. Srinivas, "Synthesis of Reactive Mass Exchange Networks," *Chem. Eng. Sci.*, **47**, 2113 (1992).
- Exner, O., "The Hammett Equation—the Present Position," *Advances in Linear Free Energy Relationships*, Chapter 1, N. Chapman, J. Shorter, eds., Plenum Press, New York (1972).
- Exner, O., *Correlation Analysis of Chemical Data*, Plenum Press, New York (1988).
- Faith, W., D. Keys, and R. Clark, *Industrial Chemicals, Third Ed.*, John Wiley and Sons, New York (1980).
- Ferguson, L., "The Synthesis of Aromatic Aldehydes," *Chem. Rev.*, **38**, 229 (1946).
- Fincham, D., N. Quirke, and D. Tildesley, "Computer Simulation of Molecular Liquid Mixtures. 1. A Diatomic Lennard-Jones Model Mixture for CO₂/C₂H₆," *J. Chem. Phys.*, **84**, 4535 (1986).
- Fisanick, W., A. Lipkus, and I. Rusinko, "Similarity Search on CAS Registry Substances. 2. 2D Structural Similarity," *J. Chem. Inf. Comp. Sci.*, **34**, 130-40 (1994).
- Fornari, T., E. Rotstein, and G. Stephanopoulos, "Studies on the Synthesis of Chemical Reaction Paths—II. Reaction Schemes with Two Degrees of Freedom," *Chem. Eng. Sci.*, **44**, 1569 (1989).
- Fourer, R., D. Gay, and B. Kernighan, *AMPL. A Modeling Language for Mathematical Programming*, Scientific Press, San Francisco (1993).
- Fredenslund, A., R. L. Jones, and J. M. Prausnitz, "Group-Contribution Estimation of Activity Coefficients in Nonideal Liquid Mixtures," *AIChE J.*, **21**, 1086 (1975).
- Gani, G., B. Nelson, and A. Fredenslund, "A Group Contribution Approach to Computer-Aided Molecular Design," *AIChE J.*, **27**, 1318 (1991).
- Ghanem, R. and P. Spanos, *Stochastic Finite Elements: A Spectral Approach*, Springer-Verlag, New York (1991).
- Geoffrion, A., *J. Opt. Theory Applic.*, **10**, 237 (1972).

- Greiner, A., W. Strettmatter, and J. Honerkamp, "Numerical Integration of Stochastic Differential Equations," *J. Statistical Phys.*, **51**, 95 (1988).
- Grossmann, I. and C. Floudas, "Active Constraint Strategy for Flexibility Analysis in Chemical Processes," *Comp. Chem. Eng.*, **11**, 675 (1987).
- Grossmann, I. and J. Santibanez, "Applications of Mixed-Integer Linear Programming in Process Synthesis," *Comp. Chem. Eng.*, **4**, 205 (1980).
- Frisch, M., G. Trucks, M. Head-Gordon, "GAUSSIAN 92," *Gaussian Inc.*, Pittsburgh, Pennsylvania (1992).
- Gasteiger, J., W. Ihlenfeldt, P. Rose, *et al.*, "Computer-Assisted Reaction Prediction and Design," *Analytica Chimica Acta.*, **235**, 65 (1990).
- Govind, R., "Automated Reaction Path Synthesis," *M. S. Thesis*, Dept. Chemical Eng., Carnegie Mellon University (1977).
- Govind, R. and G. Powers, "A Chemical Engineering View of Reaction Path Synthesis," *Computer-Assisted Organic Synthesis*, pp 81-96, W. Wipke, W. Howe, Eds., American Chemical Society, Washington, D. C. (1977).
- Govind, R. and G. Powers, "Studies in Reaction Path Synthesis," *AIChE J.*, **27**, 429 (1981).
- Grigoras, S., "A New Semi-empirical Method for the Calculation of the Partition Coefficient," Ph.D. Thesis, University of Illinois at Chicago (1985).
- Grigoras, S., "A Structural Approach to Calculate Physical Properties of Pure Organic Substances: The Critical Temperature, Critical Volume and Related Properties," *J. Computational Chem.*, **11**, 493 (1990).
- Groggins, P., "Nitration," *Unit Processes in Organic Synthesis, Fifth Ed.*, pp 59-77, P. H. Groggins, ed., McGraw-Hill, New York (1958).
- Grossmann, I., "MINLP Optimization Strategies and Algorithms for Process Synthesis," *Foundations of Computer-Aided Process Design*, pp 105-132, J. Siirola, I. Grossmann, G. Stephanopoulos, Eds., Elsevier Publishing Company, New York (1989).
- Hagedorn, F., "Nitriles," *Ullmann's Encyclopedia of Industrial Chemistry, Fifth Ed., Vol A17*, pp 370-376, B. Elvers, S. Hawkins, G. Schulz, Eds., VCH, Weinheim (1985).
- Halemane, K. and I. Grossmann, "Optimal Process Design Under Uncertainty," *AIChE J.*, **29**, 425 (1983).
- Hegarty, A. F., "Kinetics and Mechanisms of Reactions Involving Diazonium and Diazo Groups," *The Chemistry of Diazonium and Diazo Groups*, pp 511-591, S. Patai, ed., John Wiley and Sons, New York (1978).
- Hehre, W., L. Radom, and J. Pople, "Molecular Orbital Theory of the Electronic Structure of Organic Compounds. XII. Conformations, Stabilities, and Charge Distributions in Monosubstituted Benzenes," *J. Am. Chem. Soc.*, **94**, 1496 (1972).
- Hehre, W., L. Radom, P. v.R. Schleyer, *et al.*, *Ab Initio Molecular Orbital Theory*, John Wiley and Sons, New York (1986).
- Hendrickson, J. "A Systematic Characterization of Structures and Reactions for Use in Organic Synthesis," *J. Am. Chem. Soc.*, **93**, 6847 (1971).
- Hendrickson, J. "Synthesis Design for Substituted Aromatics," *J. Am. Chem. Soc.*, **93**, 6854 (1971).
- Hendrickson, J., "Systematic Synthesis Design. 6. Yield Analysis and Convergence," *J. Am. Chem. Soc.*, **99**, 5439 (1977).
- Hendrickson, J., "Approaching the Logic of Synthesis Design," *Acc. Chem. Res.*, **19**, 274 (1986).
- Hendrickson, J., "The SYNGEN Approach to Synthesis Design," *Anal. Chim. Acta.*, **235**, 103 (1990).

- Hendrickson, J., D. Grier, and A. Toczko, "A Logic-Based Program for Synthesis Design," *J. Am. Chem. Soc.*, **107**, 5228 (1985).
- Hendrickson, J. and C. Parks, "A Program for the Forward Generation of Synthetic Routes," *J. Chem. Inf. Comput. Sci.*, **32**, 209 (1992).
- Herges, R., "Reaction Planning," *Chemical Structures*, pp 385-398, W. Warr, Ed., Springer, Berlin (1988).
- Herges, R. and C. Hoock, "Reaction Planning: Computer-Aided Discovery of a Novel Chemical Reaction," *Science*, **255**, 711 (1992).
- Hertel, H., "Diazo Compounds and Diazo Reactions," *Ullmann's Encyclopedia of Industrial Chemistry, Fifth Ed., Vol A8*, pp 505-522, B. Elvers, S. Hawkins, G. Schuly, Eds., VCH, Weinheim (1985).
- Hogget, J.; R. Moodie *et al.*, *Nitration and Aromatic Reactivity*, University Press, Cambridge (1971).
- Holmes, C. and J. London, "The Mobility of Groups in Certain Benzonitriles," *J. Chem. Soc.*, 1521 (1940).
- Illman, D., "'Green' Technology Presents Challenge to Chemists," *Chem. and Eng. News*, 26 (September 6, 1993).
- Israelachvili, J., *Intermolecular and Surface Forces*, Academic Press Limited, San Diego, California (1992).
- Jaffé, H., "Correlation of Hammett's σ -Values with Electron Densities Calculated by Molecular Orbital Theory," *J. Chem. Phys.*, **20**, 279 (1952).
- Jobak, K. and G. Stephanopoulos, "Designing Molecules Possessing Desired Physical Property Values," *Foundations of Computer-Aided Process Design*, pp 363-387, J. Siirola, I. Grossmann, G. Stephanopoulos, Eds., Elsevier Publishing Company, New York (1989).
- Kaufman, N. "The Ongoing Greening of Polaroid Chemical and Polymer Synthesis Plants," *Compliance Through Waste Minimization*, New England Water Pollution Prevention Association (1994).
- Kaufman, J. and V. Drago, "Direct Access to Material Properties for Modeling and Simulation," *Modelling and Simulation in Mat. Sci. and Eng.*, **1**, 335-347 (1993).
- Kerestecioglu, A., *General Option/Security Price Evaluation Language Technical Manual, Version 1.2*, Atlantic Portfolio Analytics and Management, Orlando, Florida (1992).
- Klamt, A. and G. Schüürmann, "COSMO: A New Approach to Dielectric Screening in Solvents with Explicit Expressions for the Screening Energy and its Gradient," *J. Chem. Soc. Perkin Trans. 2*, 799 (1993).
- Knight, J., "Targeting System Chemistry in Process Synthesis: A Combined Chemometric, Retrosynthetic, Mathematical Programming Approach," Ph.D. Proposal, Massachusetts Institute of Technology (1993).
- Knight, J., "An Approach to Process Integration Based on the Choice of System Chemistry," *Process Synthesis Symposium: Paper 153d, AIChE Annual Meeting*, St. Louis (1993).
- Knight, J., "Hydroxymethylation-Oxidation Synthesis of DMB; and Summary of Available Literature on Synthesis of DMB via Vilsmeier-Haack Formylation," *Polaroid Project Rept. #2*, Polaroid (1994).
- Kocis, G. and I. Grossmann, *Ind. Eng. Chem. Res.*, **26**, 1869 (1987).
- Krohn, K., K. Khanbabaee, and H. Rieger, "Titanium- or Zirconium-Catalyzed Selective Dehydrogenation of Benzyl Alcohols to Aldehydes and Ketones with *tert*-Butyl Hydroperoxide," *Chem. Ber.*, **123**, 1357 (1990).
- Kumar, V., S. Singh, and M. Garg, "RAGE—A Software for Data Reconciliation and Gross Error Detection," *Foundations of Computer-Aided Process Operations II*, Crested Butte, Colorado (1993).

- Lang, Y., L. Biegler, and I. Grossmann, "Simultaneous Optimization and Heat Integration With Process Simulators," *Comp. Chem. Eng.*, **12**, 311 (1988).
- Ligon, J. and N. Amundson, "Modeling of Fluidized Bed Reactors—VI(b) The Nonisothermal Bed with Stochastic Bubbles," *Chem. Eng. Sci.*, **36**, 661 (1980).
- Linda, P., A. Lucarelli, G. Antonio, *et al.*, "Mechanism of the Vilsmeier-Haack Reaction. III. Structural and Solvent Effects," *J. Chem. Soc. Perkin Trans. 2*, 1610 (1974).
- Luyben., M. and C. Floudas, "Analyzing the Interaction of Design and Control in a Reactor-Separator-Recycle Stream," *AIChE Annual Meeting*, St. Louis (1993).
- Levine, I., *Quantum Chemistry, Fourth Ed.*, Prentice-Hall, Englewood Cliffs, New Jersey (1991).
- Lushnikov, D. and N. Zefirov, "Fragmentation and "Feeling the Goal" in Computer-Assisted Synthesis," *J. Chem. Inf. Comput. Sci.*, **32**, 317 (1992).
- Macchietto, S. "Bridging the Gap—Integration of Design, Operations Scheduling and Control," Draft for FOCAPO II Proceedings, V. Mahalec, Ed. (1993)
- Macchietto, S, O. Odele, and O. Omatsome, "Design of Optimal Solvents for Liquid-Liquid Extraction and Gas Absorption Processes," *Trans. IChemeE.*, **68** (Part A), 233 (1990).
- Mavrovouniotis, M., G. Stephanopoulos, and G. Stephanopoulos, "Synthesis of Biochemical Production Routes," *Computers Chem. Eng.*, **16**, 605 (1992).
- MacMullin, R. B., "Distribution Reaction Products in Benzene Chlorination. Batch vs. Continuous Process Procedures," *Chem. Eng. Prog.*, **44**, 183 (1948).
- McDermott, J. B., C. Libanati, C. LaMarca, *et al.*, "Quantitative Use of Model Compound Information: Monte Carlo Simulation of Reactions in Complex Macromolecules," *Ind. Eng. Chem. Res.*, **29**, 22 (1990).
- McKelvey, J., S. Alexandratos, A. Streitwieser, *et al.*, "Correlation of STO-3G Calculated Substituent Effects on the Proton Affinity of Benzene with σ^+ Parameters," *J. Am. Chem. Soc.*, **98**, 244 (1976).
- Miller, J. *Aromatic Nucleophilic Substitution. Reaction Mechanisms In Organic Chemistry Monograph 8*, Elsevier Publishing Company, New York (1968).
- Miller, D. and J. Pekny, "Exact Solution of Large Asymmetric Traveling Salesman Problems," *Science*, **251**, 754 (1991).
- Mills, J., C. Maryanoff, K. Sorgi, *et al.*, "REACCS in the Chemical Development Environment. 2. Structure and Construction of Proprietary Databases," *J. Chem. Inf. Comp. Sci.*, **28**, 155-159 (1988).
- Moore, R., *Interval Analysis*, Prentice-Hall, Englewood Cliffs, NJ (1966).
- Morgan, S., "Use Process Integration to Improve Process Designs and the Design Process," *Chem. Eng. Prog.*, **62**, (September, 1992).
- Muller, F. and L. Caillard, "Chlorophenols," *Ullmann's Encyclopedia of Industrial Chemistry, Fifth Ed., Vol A7*, pp 1-8, B. Elvers, S. Hawkins, G. Schuly, Eds., VCH, Weinheim (1985).
- Murray, J., T. Brinck, and P. Politzer, "Partition Coefficient of Nitroaromatics Expressed in Terms of Their Molecular Surface Areas and Electrostatic Potentials," *J. Phys. Chem.*, **97**, 13807 (1993).
- Murray, J., T. Lane, T. Brinck, *et al.*, "Relationships of Critical Constants and Boiling Points to Computed Molecular Surface Properties," *J. Phys. Chem.*, **97**, 9369 (1993).
- Murtagh, B. and M. Saunders, *MINOS 5.1 User's Guide*, System Optimization Laboratory, Dept. Operations Res., Stanford University, Technical Rept. SOL 83-20R (1987).

- Naser, S., R. Fournier, "A System for the Design of an Optimal Liquid-Liquid Extractant Molecule," *Comp. Chem. Eng.*, **15**, 397 (1991).
- Nemhauser, G. and L. Wolsey, *Integer and Combinatorial Optimization*, John Wiley and Sons, New York (1988).
- Neurock, M., C. Libanati, and M. Klein, "Modelling Asphaltine Reaction Pathways: Intrinsic Chemistry," *Fundamentals of Resid Upgrading. AIChE Symposium Series, No. 273*, pp 7-14 R. Heck, T. Degnan, Eds. (1989).
- Nigam, A. and M. Klein, "A Mechanism Oriented Lumping Strategy for Heavy Hydrocarbon Pyrolysis: Imposition of Quantitative Structure-Reactivity Relationships for Pure Components," *Ind. Eng. Chem. Res.*, **32**, 1297 (1993).
- Nishida, N., G. Stephanopoulos, and A. W. Westerberg, "A Review of Process Synthesis," *AIChE J.*, **27**, 321 (1981).
- Nishino, H., K. Tsunoda, and K. Kurosawa, "Manganese(III)-Mediated Formylation of Aromatic Compounds in the Presence of Malonic Acid," *Bull. Chem. Soc. Jpn.*, **62**, 545 (1989).
- Nobel Chemicals, "5-Nitroisophthalic Acid Technologies," *Chem. and Eng. News*, 58 (October 18, 1994).
- Norman, R. and R. Taylor, *Electrophilic Substitution in Benzenoid Compounds*, Elsevier Publishing Company, New York (1965).
- Odian, G., *Principles of Polymerization*, John Wiley and Sons, New York, 1991.
- Olah, G. and S. Kuhn, *Friedel-Crafts and Related Chemistry, Vol. 2*, G. Olah, Ed., Wiley-Interscience, New York (1964).
- Paul, M. A., "Kinetics of Aromatic Nitration in Acetic Anhydride," *J. Am. Chem. Soc.*, **80**, 5329 (1958).
- Perkins, J. and R. Sargent, "SPEEDUP: A Program for Steady-State and Dynamic Optimization of Chemical Processes," *AIChE Series Symposium No. 8* (1992).
- Peters, M. S. and K. D. Timmerhaus, *Plant Design and Economics for Chemical Engineers, Third Ed.*, McGraw-Hill, New York (1980).
- ***Pople, J. and D. Beveridge, "," *J. Chem. Phys.*, **47**, 2026 (1967).
- Pople, J. and D. Beveridge, *Approximate Molecular Orbital Theory*, McGraw-Hill, New York (1970).
- Powers, G. J., R. L. Jones, G. A. Randall, *et al.*, "Optimal Strategies for the Chemical and Enzymatic Synthesis of Bihelical Deoxyribonucleic Acids," *J. Am. Chem. Soc.*, **97**, 875 (1975).
- Powers, G. and J. Mayer, "Design of a Catalytic Reactor-Separator System with Uncertainty in Catalyst Activity and Selectivity," *Ind. Eng. Chem. Process Des. Develop.*, **14**, 41 (1975).
- Press, W. H., B. P. Flannery, S. A. Teukolsky, *et al.*, *Numerical Recipes in C*, Cambridge University Press (1991).
- Prickett, S., L. Constantinou, and M. Mavrouniotis, "Computational Identification of Conjugate Paths for Estimation of Properties of Organic Compounds," *Molecular Simulation*, **11**, 205 (1993).
- Purcell, A., "The Third Dimension of Waste Minimization," *Chem. Eng. Prog.*, **5**, (June, 1989).
- Quann, R. and S. Jaffe, "Structure-Oriented Lumping: Describing the Chemistry of Complex Hydrocarbon Mixtures," *Ind. Eng. Chem. Res.*, **31**, 2483 (1992).
- Radom, L., W. Hehre, and J. Pople, "Conformations of Heats of Formation by Organic Molecules by Use of a Minimal Slater Type Basis," *J. Chem. Soc. (A)*, 2299 (1971).
- Reichardt, C., *Solvents and Solvent Effects in Organic Chemistry*, VCH, Weinheim (1990).
- Reid, R. C., J. M. Prausnitz, and B. E. Poling, *The Properties of Gases and Liquids, 4th Ed.*, McGraw-Hill, New York (1987).

- Rice, O. K., *Statistical Mechanics: Thermodynamics and Kinetics*, W. H. Freeman and Co., San Francisco (1967).
- Rippen, D., "Batch Process Systems Engineering: A Retrospective and Prospective Review," *Computers chem. Engng., Supplement*, **17**, S1 (1993).
- Rose, P. and J. Gasteiger, "Automated Derivation of Reaction Rules for the EROS 6.0 System For Reaction Prediction," *Analytica Chimica Acta.*, **235**, 65 (1990).
- Rossiter, A. P., H. Spriggs, and H. Klee, "Apply Process Integration to Waste Minimization," *Chem. Eng. Prog.*, **30**, (January, 1993).
- Salatin, T. and W. Jorgensen, "Computer-Assisted Mechanistic Evaluation of Organic Reactions. 1. Overview," *J. Org. Chem.*, **45**, 2043 (1980).
- Schmidt, S. and R. Launsby, *Understanding Industrial Designed Experiments, Third Ed.*, Air Academy Press, Colorado Springs (1992).
- Shafer, G., *Mathematical Theory of Evidence*, Princeton University Press, Princeton, NJ (1976).
- Shankaran, K. and A. Rao, "A Convenient Route for the Preparation of Aromatic Aldehydes," *Synth. Commun.*, **10**, 573 (1980).
- Shreve, R., "Amination By Reduction," *Kirk-Othmer Encyclopedia of Chemical Technology, Second Ed., Vol 2*, pp 76-99 (1963).
- Smythe, S., M. Cole, G. Martin, et al., "Process for the Chlorination of Nitroanilines with Chlorine in Acetic Acid as a Medium," *NorAm Chemical Co.*, U. S. Patent 5068443 A (November 26, 1991).
- Springer-Verlag, *Synergisms in Chemical Information*, Spinger New Media, New York (1992).
- Staverman, A., "The Entropy of High Polymer Solutions. Generalization of Formulae," *Rec. Trav. Chim. Pays-bas*, **69**, 163 (1950).
- Steele, K., Personal Correspondence, DowElanco (1993).
- Stewart, J. and Fujitsu Ltd., MOPAC 93 Manual, *Fujitsu Systems Business of America, Inc.*, Santa Clara, CA (1993).
- Stinson, S. "Custom Chemicals," *Chem. and Eng. News*, **34** (February 8, 1993).
- Stock, L. M., *Aromatic Substitution Reactions*, Prentice-Hall, Englewood Cliffs, New Jersey (1968).
- Stock, L. and H. Brown, H. "A Quantitative Treatment of Directive Effects In Aromatic Substitution," *Advances in Physical Organic Chemistry*; pp 35-154, V. Gold, ed., Academic Press, New York (1963).
- Straub, D. and I. Grossmann, "Integrated Stochastic Metric of Flexibility for Systems with Discrete State and Continuous Parameter Uncertainties," *Computers chem. Engng.*, **14**, 967 (1990).
- Straub, D. and I. Grossmann, "Evaluation and Optimization of Stochastic Flexibility in Multiproduct Batch Plants," *Computers chem. Engng.*, **16**, 69 (1992).
- Streitwieser, A. and C. Heathcock, *Introduction to Organic Chemistry*, Macmillan Publishing Co., New York (1981).
- Swaney, R. and I. Grossmann, "An Index for Operational Flexibility in Chemical Process Design. Part I: Formulation and Theory," *AIChE J.*, **31**, 621 (1985).
- Swaney, R. and I. Grossmann, "An Index for Operational Flexibility in Chemical Process Design. Part II: Computational Algorithms," *AIChE J.*, **31**, 631 (1985).
- Szabo, A. S. and N. S. Ostlund, *Modern Quantum Chemistry, First Ed., Revised*, McGraw-Hill, New York (1989).
- Taha, H., *Operations Research. An Introduction. Fourth Ed.*, MacMillan Publishing, New York (1987).

- Tatang, M., *Ph.D. Thesis*, Massachusetts Institute of Technology (1994).
- Taylor, R., *Electrophilic Aromatic Substitution*, John Wiley and Sons, New York (1990).
- Traina, M., R. Miller, and S. Masri, "Orthogonal Decomposition and Transmission of Nonstationary Random Processes," *Probabilistic Eng. Mech.*, **1**, 136 (1986).
- Trost, B., "The Atom Economy—The Search for Synthetic Efficiency," *Science*, **254**, 1471 (1991).
- Trucks, G. W., M. Head-Gordon, P. M. W. Gill, *et al.*, Gaussian, Inc., Pittsburgh, Pennsylvania (1992).
- Vasantharajan, S., J. Viswanathan, and L. Biegler, "Reduced Successive Quadratic Programming Implementation for Large-Scale Optimization Problems with Smaller Degrees of Freedom," *Comp. Chem. Eng.*, **14**, 907 (1990).
- Viswanathan, J. and I. Grossmann, *Comp. Chem. Eng.*, **14**, 769 (1990).
- Vogt, P. F., "Amines, Aromatic," *Ullmann's Encyclopedia of Industrial Chemistry, Fifth Ed., Vol A2*, pp 37-55, B. Elvers, S. Hawkins, G. Schuly, eds., VCH, Weinheim (1985).
- Voudouris, V. and I. Grossmann, "MILP Model for Scheduling and Design of a Special Class of Multipurpose Batch Plants," *Computer Integrated Manufacturing Symposium: Paper 144d, AIChE Annual Meeting*, St. Louis (1993).
- Werner, J. and P. Groggins, "Amination By Reduction," *Unit Processes in Organic Synthesis, Fifth Ed.*, pp 129-203, P. H. Groggins, ed., McGraw-Hill, New York (1958).
- Wiegandt, H. F. and P. R. Lantos, "Improved Yields of p-Dichlorobenzene," *Ind. Eng. Chem.*, **43**, 2167 (1951).
- Weinstock, R., *Calculus of Variations*, Dover Publications, New York (1974).
- Wozniakowski, H., "Average Case Complexity of Multivariate Integration" *Bull. Am. Math. Soc.*, **24**, 185 (1991).
- Yuta, K. and P. Jurs, "Computer-Assisted Structure-Activity Studies of Chemical Carcinogens. Aromatic Amines," *J. Med. Chem.*, **24**, 241 (1981).
- Zadeh, L., "Fuzzy Sets," *Information and Control*, **8**, 338 (1965).
- Zeegers-Huyskens, Th. and P. Huyskens, "Intermolecular Forces," *Intermolecular Forces*, pp 2-30, P. Huyskens, W. Luck *et al.*, eds., Springer-Verlag, New York (1991).
- Zefirov, N. and S. Trach, "Symbolic Equations and Their Applications to Reaction Design," *Anal. Chim. Acta*, **235**, 115 (1990).