OPTIMAL DESIGN AND OPERATION OF BATCH PROCESSES

by

Michael D. Barrera

B.S. Ch.E. Virginia Polytechnic Institute and State University
(1984)

M.S. C.E.P. Massachusetts Institute of Technology
(1985)

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

June 1990

Signature of Author _____
Department of Chemical Engineering
March 29, 1990

Certified by _____
Lawrence B. Evans
Thesis Supervisor

Accepted by _____
William M. Deen
Chairman, Committee for Graduate Students

OPTIMAL DESIGN AND OPERATION OF BATCH PROCESSES

by

Michael D. Barrera

Submitted to the Department of Chemical Engineering
on March 29, 1990 in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy in Chemical Engineering

ABSTRACT

The increased importance of high value-added specialty chemicals has stimulated interest in the development of better design and optimization methods for batch processes. These processes have a number of particular aspects (time-dependent behavior, discrete processing and structural alternatives) which make the development of optimal designs quite difficult. Five sets of decisions must be made to design a batch process. Because the entire problem is so complex, previous workers have focused on smaller, more manageable parts of the overall problem. Much of the previous work has dealt with the optimal sizing of equipment units in order to produce a set of products in a new multiproduct batch plant with minimum capital investment. Processing conditions have generally been assumed given in the form of the product recipes and not subject to change.

This thesis focuses on the design and operation of batch processes and represents a first attempt to consider process performance issues and equipment sizing decisions together in an optimization framework. Complexities introduced by using existing equipment are also included. The optimization problem consists of selecting the equipment units to use at each stage in the process and choosing values for all process operating conditions and operating times in order to optimize a suitable objective function.

A problem formulation for the optimal design of a new batch process is developed. This formulation incorporates the effects of the overall multipurpose plant environment on the design of a single new process by allocating fixed costs through the use of equipment usage charges. These charges represent the opportunity cost of allocating scarce plant resources to one product rather than another. A decomposition strategy is proposed for solving the optimization problem. By partitioning the decision variables into two groups, simpler subproblems are generated.

The Performance Subproblem involves optimizing the continuous variables that represent the processing conditions and operation times for a process with fixed structure. Generic performance trade-offs involving processing intensity and the distribution of performance load are identified. The problem is formulated as a nonlinear programming problem (NLP) and solved using a successive quadratic programming algorithm. Results

are reported for a series of test problems to illustrate the basic elements of the solution approach.

The Structure Subproblem involves assigning known equipment units to stage locations for a process with fixed performance in order to minimize the total equipment usage charges. The combinatorial optimization problem can be formulated as a mixed integer nonlinear programming problem (MINLP) and solved using mathematical programming methods such as the Outer Approximation, Equality Relaxation method. To circumvent potentially large solution times, an approximate solution strategy based on local search techniques is developed. This approximate method generates near-optimal solutions and requires one to three orders of magnitude less computational effort than exact techniques.

Solving the combined problem (process performance and equipment assignment) requires coordinating the solution methods for the two subproblems to obtain an overall optimum. Two alternative nesting strategies are considered as possible solution methods. Embedding the performance subproblem inside the structure subproblem is preferred for convergence reasons, but this approach becomes time consuming for larger problems. Although placing the performance optimization in the outer loop can reduce the computational load, efficient algortihms for solving the resulting non-unimodal, discontinuous objective function do not exist.

A case study design problem based on a process development project from industry is solved to determine the effects of applying the solution strategies developed to a "real world" problem. Although the problem data are modified, the complexities and types of trade-offs present in the actual project are retained. An evolutionary strategy is employed to avoid excessive computational requirements that would arise from embedding detailed process models inside a mathematical programming optimization routine. Key performance trade-offs are identified and explored in a systematic way using the process simulation model as a design tool. The approximate solution strategies work well on the equipment assignment aspects of the problem. This case study illustrates the significant benefits that can be obtained by considering both process performance and equipment assignment together at the design stage. Optimizing either performance or structure independently leads to more costly process designs.

Areas for future research include the development of computational speed-ups for the performance subproblem. The use of short-cut design models and optimization schemes that employ reduced models (e.g., simultaneous modular approach) are potential approaches that could lower the high computational requirements expected when industrially sized problems are considered. In addition, the approximate methods based on local search could be applied to a number of other discrete optimization problems that arise during batch process design. Future work would be required to determine the problem dependent performance characteristics of this type of approximate approach for each new problem type.

Thesis Supervisor: Dr. Lawrence B. Evans
Title: Professor of Chemical Engineering

ACKNOWLEDGEMENTS

I would like to thank my thesis advisor, Professor Lawrence B. Evans, for his patient support and guidance in helping me see this project through to the end. He has always been available to help put the big picture back into focus when I have become mired in a morass of seemingly endless details. I would also like to thank Dr. Charles Cooney, Dr. Mark Kramer, and Dr. Michael Mohr for serving on my thesis committee.

Financial support for this thesis has been provided primarily by a National Science Foundation Graduate Fellowship. Additional financial support has come from the National Science Foundation under cooperative agreement #CDR-88-03014 and from the United States Department of Energy. This funding has been greatly appreciated.

Sometimes, when completing a project like this one, it is not "what you know," but "who you know" that counts. Fortunately, I have had the privilege to associate with some great individuals during my N years at MIT. I would like to thank my office-mates for all their contributions that helped make this thesis a reality. Reiner Musier has been a great source of wisdom, especially on combinatorial optimization. Special recognition goes to Suresh Sundaram for his collaboration on the case study problem, which has been used in the Batch Process Design Module of the ICE Course for the past two years. Suresh's expertise with batch distillation has been particularly helpful. I would also like to thank Murali Viswanathan and Suresh for their efforts as system managers for our group's Microvax computer.

For all the people who have made life interesting and enlightening for the past few years (JC, RR, LC, MA, HDF), I will always remember my time at MIT with special fondness. I'd also like to thank everyone that captained an IM team and Bob Kiss for their time and efforts in making the IM program in the department the success that it is today.

Finally, I'd like to acknowledge my fiance Denise, without whom this thesis would never have been completed "on time".

DEDICATION

To my parents, Kenneth and Alice Barrera, and my fiance Denise.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Growth of Batch Processes

Recently, there has been an increased emphasis by the Chemical Processing Industries (CPI) on the production of higher value-added specialty products. Some of the most profitable segments of the CPI today are focusing on making these higher margin materials as opposed to more traditional bulk commodity products. This move reflects the growing market-driven environment in which chemical producers find themselves. Satisfying customer requirements with specialized product grades and producing materials to exploit niche markets are increasing concerns.

Bulk, commodity chemicals are typically produced by the CPI in continuous operations in dedicated plants. This is not the case, however, for specialty products, which tend to have smaller markets. The lower annual production requirements associated with these products do not justify the construction of dedicated, continuous plants because each plant would operate with an uneconomically small processing rate.

Batch processes offer a number of distinct advantages over continuous processes for companies producing low volume products. First, by using shared, standardized equipment items, a single multiproduct facility can be used to make a number of products and generate a cost savings based on economy of scale. The sharing of equipment units among a number of products allows for a more efficient employment of resources. The annual production amounts for the various products might be produced in a series of relatively short campaigns. Alternatively, individual batches of the different products could be scheduled on available units as needed to satisfy customer demand.

These two potential operating schemes represent a second advantage of batch processes, namely increased flexibility. Batch plants offer greater flexibility because they can better handle variations in feedstocks and product specifications. In addition, they can shift production among various products or product grades in response to changing market

demands. Finally, many specialty and biochemical products cannot be made continuously for process-related reasons and therefore must be produced batchwise.

1.2 Need for Good Design Methods

The growing economic importance of specialty products has increased the desire for methods that would allow the more rapid design of lower cost processes. For example, if faster design methods could shorten the time for process development, a company might realize tremendous profits by beating its competitors to a new market. Opportunities to improve the operating efficiencies of existing processes also exist. By developing more rigorous methods of analysis to bring to bear on the problem, inefficient designs based on tradition and intuition can be replaced by more efficient, cost effective processes.

The design of continuous units is fairly well understood; most undergraduate chemical engineering students are introduced to general design procedures for use with continuous systems in their course work. These systems are described by sets of nonlinear algebraic equations for the mass and energy balances and the physical property relationships. The design of heat exchangers, tray to tray calculations in continuous distillation, and the sizing of pumps are fairly common examples found in the undergraduate curriculum.

However, general design procedures for batch and semi-continuous operations are not nearly so well developed. Batch processes involve cases where the governing equations are differential as well as algebraic. The typical undergraduate chemical engineering student receives much less formal training in the more difficult batch design problem, usually being exposed to only batch reactor kinetics and the general concepts behind batch distillation. The integration of batch units into a complete flowsheet is usually never addressed. The presence of a two day AIChE short course on the design of batch processing plants attests to the general lack of formal training in batch process design. In addition, batch process design problems often include combinatorial aspects that make solving the problem both conceptually difficult and computationally expensive.

The recent economic developments mentioned above, coupled with the general lack of sophisticated methods for cases involving batch and semi-continuous units, have stimulated increased interest in the field of batch process design. Recent research has attempted to develop better systematic approaches for attacking the batch design problem. Because the number of feasible design alternatives for batch processes is quite large,

researchers are currently developing methods for designing "optimal" batch processes. These approaches often make use of computer-based solution techniques.

1.3 Computer Aids

Design methods for continuous processes have reached an advanced state of development, and computer tools for simulating and optimizing processes are now used routinely. This is not the case, however, for batch processes. Parakrama (1985) conducted a survey of industrial batch processes that highlighted the needs for computer aids in the areas of design and operation. To optimize the design of batch plants on a commercial scale, computing tools will be needed with capabilities comparable to those available for continuous processes today. The evolution of these computer tools for batch operations is expected to parallel the historical development of tools for continuous processes.

Digital computers were first employed by chemical engineers to model single continuous process units. These models were custom-made to solve specific problems. As more unit models became available, simple packages were developed that contained some standard unit blocks. FLOWTRAN (Rosen and Pauls, 1977) is an example of this type of package. As the benefits of a "systems modeling" approach became apparent, larger general purpose process simulators were developed. These software packages contained physical property data banks, standard unit operation blocks, and numerical packages to solve systems of simultaneous nonlinear algebraic equations. Commercial packages such as PROCESS (Brannock et al., 1979), DESIGN/2000 (ChemShare Corp., 1979), and ASPEN PLUS (Aspen Technology, Inc., 1988) are current examples. These commercial simulators were quick to add capabilities for equipment sizing, costing, and process economics. The addition of costing functions allowed case studies to be done to try out process modifications and determine the economic ramifications of proposed changes. Commercial simulators have also added process optimization capabilities to allow trade-offs to be studied in a systematic fashion.

Batch process simulation is in its infancy at the present time in terms of being able to rigorously model batch operations. The need and desire for computer tools in industry was clearly evident in the results of Parakrama's 1985 survey of 99 industrial batch processes in the United Kingdom. The two areas that topped the industrial wish list were tools for scheduling process production and optimizing process operating conditions. In

order to optimize operating conditions, a process model that simulates the system behavior must be developed.

The first step in modeling batch processes is the simulation of individual unit operations. Modeling single batch units is somewhat more complicated than continuous ones because batch processes are inherently non-steady state. Thus, the describing equations are differential as well as algebraic and require the use of numerical integration packages. A number of attempts have been made to develop software to assist with batch process design. MULTIBATCH was developed by Sparrow et al. (1974) to model the behavior of an entire batch process and to incorporate some systematic design procedures. The models for this package were simple split fraction models or user-supplied FORTRAN subroutines. Hainsworth (1984) described the use of a simulation package called HOCUS which could be used to assist in batch process design. The applications in his examples concentrated on the modeling and design of polymer production facilities.

More recently a commercial simulator, BATCHES, designed especially for batch process systems, has been introduced (Joglekar et al., 1984; Clark and Kuriyan, 1989). This simulator provides the user with a set of general process tasks which can be combined into a process model. The software allows relatively "easy" simulation of a batch system. BATCHES for batch systems might correspond roughly to an early version of FLOWTRAN for continuous ones. Further improvements, such as general economics capabilities and process optimization, have not been implemented as yet. Noting the similarities in the development of continuous and batch computer aids, it is expected that these capabilities will be added in the future.

1.4 Potential Benefits

The use of rigorous process models in a general purpose simulation package with robust optimization capabilities for the design of multipurpose batch plants is certainly many years away at the present time. However, the trend towards a more-detailed modeling approach is already underway as shown by the development of software such as BATCHES. Faster, more powerful computers will make the simulation of unsteady-state operations easier to handle.

There are many potential benefits to applying a more-detailed modeling approach. "What if" questions can be explored, and sensitivity studies can be used to more effectively identify pilot plant studies that should be conducted. Good process models will shorten the

time required to bring a new product to market, result in more efficient, lower cost processes, and allow various process alternatives (flowsheets) to be compared more accurately at the design stage. Using a systems approach to optimize the overall process will allow various alternatives to be compared at their best. Finally, better process modeling is a requirement for including good estimates of operating costs in any overall process optimization.

The general problem of designing an "optimal" batch process is an exceedingly difficult one. There are many aspects that must be considered in the optimization analysis. Most researchers have tackled smaller pieces of the larger problem. These "easier" subproblems are complicated and difficult problems in their own right. One assumption used by most workers during the optimal design of batch processes is that the process behavior is known and fixed in advance by the process recipe. This assumption allows very simple models to be used to represent process performance. Fixing the process performance reduces the computational effort required to solve the various optimization subproblems that have been proposed. However, this assumption also reduces the scope for improving the overall batch process design.

Workers using more-detailed models have shown that there are benefits to be gained by exploring variations in processing conditions and operating strategies (Wilson, 1987; Young and Reklaitis, 1989). However, powerful optimization techniques that can handle many of the aspects of batch processes have not yet been incorporated into general purpose modeling tools such as the one described above. The eventual goal in the batch process design area is the bringing together of both detailed modeling tools and sophisticated optimization techniques in an overall framework for the optimal design of batch processes.

1.5 Thesis Objectives

This thesis focuses on the design and operation of batch chemical processing systems and represents a first attempt at bringing process performance issues into the optimization picture. In addition, complexities introduced by the use of existing equipment units will be examined. The "optimal design" problem considered here consists of selecting the equipment units to use at each stage in the process and determining the appropriate values for operating times and process operating variables so that an economic objective function is optimized.

ne solution procedures for this problem incorporate quantitative performance models for the process units so that overall process performance and equipment sizing can be considered simultaneously in the optimization analysis. Thus, economic trade-offs caused by varying unit operation times, process operating variables, and equipment sizes can be investigated systematically for the first time. An important goal is the development of an understanding of the underlying nature and fundamental structure of this particular problem. The potential benefits of including these more-detailed performance models in the optimal design of batch chemical processing systems are also demonstrated.

1.6 Overview of Thesis

An introduction to the important aspects of batch processes and a review of the literature dealing with batch process design are covered in Chapter 2. This section summarizes a number of basic terms that are used in the area and highlights the design subproblems that have been addressed to date. This chapter provides the context in which to place the work to be described in later chapters of this thesis.

Chapter 3 describes the specific problem formulation to be addressed in this thesis. This version of the design problem is actually a subproblem of the general optimal batch design problem. This particular formulation of the optimal batch design problem emphasizes the importance of both sizing and process operating considerations. In addition, the use of existing equipment is stressed. Two smaller problems, the "performance" and "structure" subproblems, are also defined.

Chapter 4 deals with the performance subproblem. This subproblem involves choosing process operating variable values and times for a process with fixed structure (i.e., units with known sizes) in order to optimize a suitable objective function. Two types of fundamental performance trade-offs are identified and described. The optimization problem is formulated as a nonlinear programming problem (NLP). A mathematical programming solution technique is proposed for small problems. Example problems are solved to illustrate the method and show the benefits of optimizing performance for batch processes.

Chapter 5 deals with the structure subproblem. This subproblem involves assigning known equipment units to stage locations for a process with fixed performance in order to minimize the total equipment usage charges. Various types of structural trade-offs are reviewed. The optimization problem is combinatorial and can be formulated as a mixed

integer nonlinear programming problem (MINLP). Although this MINLP can be solved by mathematical programming methods, an approximate technique based on local search methods has been developed that obtains near-optimal solutions very quickly. Example problems are solved to illustrate the performance of this approximate solution approach.

Chapter 6 deals with the more general problem formulated in Chapter 3, which involves optimizing both process performance and process structure. Interactions between performance and structure are demonstrated, and two alternative nesting arrangements for a decomposition solution strategy are studied. These two embedding strategies are tested on a pair of problems to identify the key issues in developing a methodology for this combined problem.

Chapter 7 describes an example problem based on a process development project from industry. Although the problem has been disguised in a number of ways, it retains many of the complexities and trade-offs present in the actual project. This detailed example problem serves as the vehicle for demonstrating the benefits of exploring performance and structural trade-offs in a systematic fashion.

Chapter 8 contains some brief concluding remarks. A discussion of promising directions for future work is also included. An Appendix following the body of the thesis contains detailed descriptions of the models used for example problems and the case study problem described in Chapter 7.

Chapter 2

LITERATURE REVIEW

2.1 Aspects of Batch Processes

Batch processes have a number of characteristics which make them different from the continuous processes with which chemical engineers are more familiar. These differences introduce additional complexities during the design of new batch processes. The goals of this chapter are to provide an introduction to the batch process design area and review some of the more important work that has been done in the field. There are a number of key terms and definitions that are peculiar to the batch processes. These terms are defined so that they can be used later when describing the previous work in the field. The final purpose of this chapter is to provide the context in which the work in this thesis fits.

There are four aspects of batch processes which will be described before beginning the literature review. These four aspects are: (1) the element of time, (2) the variety of structural options, (3) the range of operating modes, and (4) the concept of the bottleneck. These characteristics introduce complexities that make the design of batch systems more "difficult" than the design of continuous plants.

2.1.1 Element of Time

For the most part, continuous processes can be described by sets of nonlinear algebraic equations. The equations mathematically describe the mass and energy balances and the appropriate physical property relationships for the system. These processes operate at steady state (assuming the control system is working properly), and each unit can be characterized by its processing rate.

However, time plays an important role in batch (non-continuous) processes. Batch operations often generally consist of a series of operating steps or tasks. Each of these processing tasks will have some time requirement associated with it. State changes occur when the end-points of operating steps are reached. These state events occur at discrete

points in time and demonstrate one way in which batch processing systems are different from continuous ones.

Because these systems do not operate at a fixed steady state, the describing equations are both algebraic and differential. The modeling and the simulation of these systems become more difficult because the differential equations must be integrated. In addition, units cannot be characterized by their processing rates alone. The average processing rate for a unit is a function of the amount of material processed during an operating cycle and the length of this cycle. In general, the processing rate will not be constant over time. The cycle time becomes an important design variable.

Rippin (1983a) defined "performance" as the extent to which a task is carried out. This performance could be measured by comparing the state of an outlet stream to that of an inlet stream or to some external specification. Reactor conversion and the fractional recovery for a separation are two examples of possible performance measures. The performance of a unit will depend on the operating conditions during the processing cycle as well as the cycle time.

The time dependent nature of batch processes means that process units may operate in a number of ways. A unit may be assigned a particular type depending on its operating characteristics. Non-continuous systems can contain a variety of unit types, including: (1) continuous, (2) semi-continuous, (3) batch, and (4) semi-batch.

Continuous and semi-continuous units are characterized by their processing rates. Both continuous and semi-continuous units operate at steady state while they are running. Continuous units operate all the time except for occasional shutdowns for maintenance. Because continuous units are always operating, they must be preceded and followed by intermediate storage tanks in order to interact with the other units in a non-continuous system. Semi-continuous units, on the other hand, are shut down periodically during normal operations. Fluctuations in performance that occur during start-up and shut-down are assumed to be negligible for a semi-continuous unit. An example of a semi-continuous unit is a feed pump to a batch reactor. The pump operates only during vessel filling and is shut down for the remaining part of the operating cycle.

Batch and semi-batch units are characterized by their size or volume. Batch units have distinct filling, processing, draining, and preparation (e.g., cleaning) segments to their operating cycles. No mass crosses the unit boundaries during the processing step for

a batch unit. Thus, the average processing rate is determined by dividing the batch size by the total unit cycle time. Semi-batch units are the same as batch units except that mass may cross the system boundaries during the processing step. These units are the most complicated to model. A fed-batch fermentor operates as a semi-batch unit because nutrients are fed during the processing part of the cycle.

2.1.2 Structural Aspects

As mentioned above, a general batch process consists of a series of operating steps to be carried out in a specified order. These processing tasks comprise the recipe for the product. There are a number of structural decisions which must be made in order to convert these recipe tasks into a series of process operations in the plant. First, the number of processing stages must be decided upon. Then, the tasks must be assigned to the stages. In addition, the number of parallel units at each stage must be determined. These equipment units also need to be sized. Finally, storage can be placed between processing stages to increase the average production rate or increase the flexibility of operation. Each of these structural aspects (task to stage assignment, parallel units, and intermediate storage) will be described briefly.

Figure 2-1 shows a recipe for a simple batch process which consists of five tasks. One possible flowsheet could have a separate stage for each task, or five stages in total. This situation represents the totally split case and is shown in Figure 2-2. However, in many cases a number of tasks can be carried out in a single stage. Suppose all the tasks could be carried out in a single stage, as shown in Figure 2-3. This structure represents the totally merged case. Obviously, several choices for the number of stages exist between the limits of totally split and totally merged.

For any given number of stages, the total number of possible task to stage assignments is given by:

$$\text{Total Number} = \sum_{n_S=1}^{n_T} C_{n_S-1}^{n_T-1} \qquad (2\text{-}1)$$

This expression assumes that any combination of adjacent tasks can be carried out in a single stage. This formula essentially gives the number of ways of inserting (n_S-1) lines between n_T items, where n_S is the number of stages and n_T is the number of tasks. Two possible task to stage assignments for the case when there are five tasks and three stages are shown in Figure 2-4.

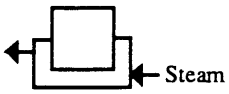
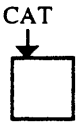
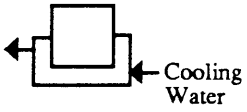
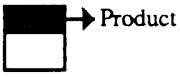
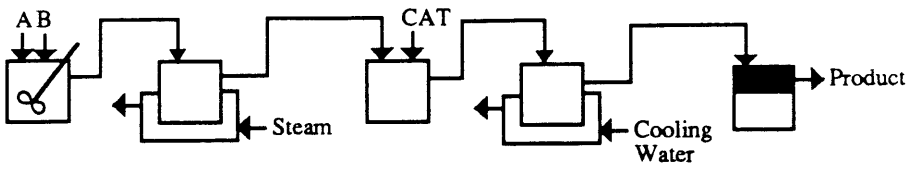
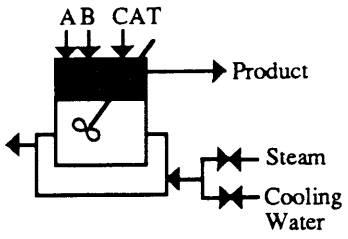
Task #	Task	Icon
1	Mix A and B	
2	Heat	
3	Add Catalyst and React	
4	Cool	
5	Decant Liquid Phase Containing Product	

FIGURE 2-1. RECIPE FOR A SIMPLE BATCH PROCESS



Stage	**1**	**2**	**3**	**4**	**5**
Task	**1**	**2**	**3**	**4**	**5**

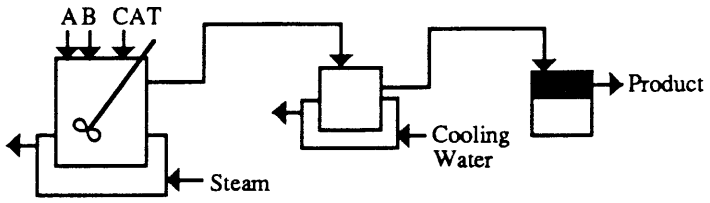
FIGURE 2-2. TOTALLY SPLIT TASK TO STAGE ASSIGNMENT



Stage	**1**
Task	**1, 2, 3, 4, 5**

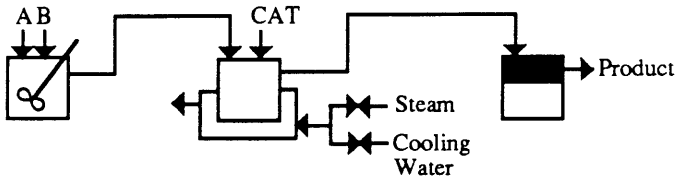
FIGURE 2-3. TOTALLY MERGED TASK TO STAGE ASSIGNMENT

Case I



Stage	1	2	3
Task	1, 2, 3	4	5

Case II



Stage	1	2	3
Task	1	2, 3, 4	5

FIGURE 2-4. POSSIBLE TASK TO STAGE ASSIGNMENTS

Batch systems often contain parallel units. Parallel units are used for two reasons. The first case occurs when there is some restriction on the maximum size of a particular unit. Parallel units are then operated in phase to increase the size of the batch that is sent to the next stage. In the second case, parallel units are used to decrease the effective cycle time of a stage by operating the parallel units out of phase. Using parallel units in this fashion allows the equipment at other stages to be utilized more fully. Parallel units are usually assumed to be identical in most solution methods described in the literature. In practice, the units could be non-identical. The inclusion of parallel units in the optimal design problem introduces combinatorial aspects to the problem because the number of parallel units must be an integer.

Batch processes can often use intermediate storage to improve operations. Intermediate storage is not usually an important design consideration in continuous plants. Interstage storage tanks are usually included only for start-up and shut-down purposes and for safety considerations. However, for the non-continuous plant, storage has an important impact on normal operations. Intermediate storage can be used to decouple the operation of adjacent stages. This decoupling increases the flexibility of the process and can lead to higher average processing rates.

There are a number of possible storage policies that are discussed in the literature. Four types are described below.

(1) Unlimited Intermediate Storage (UIS). This case corresponds to a situation where the interstage storage is effectively infinite in size. The size of the storage does not impose any constraints on the operation of the process.

(2) Finite Intermediate Storage (FIS). In this case the storage size is bounded. The limitation on storage size might impact on the operation of the system. This case is probably the most common situation in practice.

(3) No Intermediate Storage (NIS). In this case there is no storage between stages. The upstream unit can only discharge its batch when the downstream unit is available. Thus, the upstream unit may have to serve as its own storage tank while the downstream unit finishes the previous batch. While waiting, the upstream unit is prevented from starting the next batch. An NIS case might arise when successive batches cannot be mixed. For example,

batches of pharmaceuticals might be processed with NIS for purity and quality control reasons.

(4) Zero Wait (ZW). In this case, the batch in the upstream stage cannot wait for a downstream unit to become ready. Thus, the upstream unit cannot serve as its own storage tank. The upstream unit must not start processing the current batch until it is certain that a downstream unit will be ready when the upstream unit finishes processing. An unstable reaction mixture that must be quenched immediately would be an example of a ZW case.

2.1.3 Operating Modes

The time dependent nature of batch processes increases the number of operating options. Batch plants can be classified by the number of products that they produce, the way individual batches are assigned to stages of equipment, and the way these assignments are made over time. This assignment of batches to stages over time is called scheduling. Gantt Charts can be used to pictorially represent the scheduling of batches to stages. Typically, the stages are listed along the y axis, and time is shown on the x axis. Figure 2-5 shows a Gantt Chart for a two stage process producing two batches each of products A and B.

Continuous processing plants are typically dedicated systems, i.e., the process units are used to produce one major product. However, there are number of categories into which batch processing plants fall. These are shown in Table 2-1 as a function of the number of products and the time horizon over which scheduling decisions are made. A single product batch plant is also dedicated to the production of a single main product. In a short term scheduling environment, batches are made to order and immediately shipped. The plant would be idle between orders. In a longer term scheduling environment, repeated batches are produced in a campaign. Product shipments would then come from inventory.

There are a number of plant types when multiple products are produced. The differences arise from the way in which products are assigned to equipment units. In a flowshop, all the products follow the same processing route through a series of stages. In a jobshop, products do not necessarily follow the same processing route. The jobshop represents the more general case.

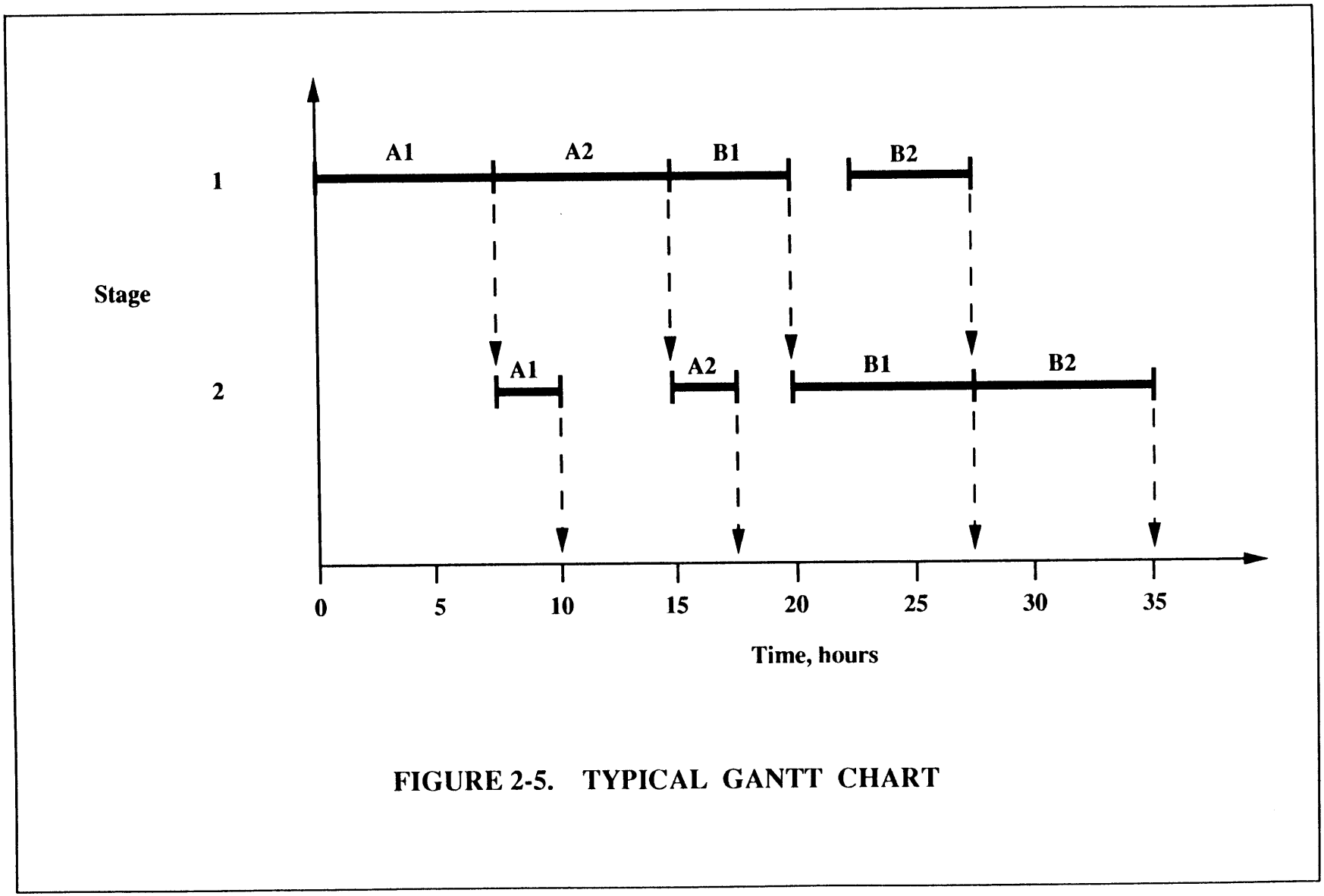


FIGURE 2-5. TYPICAL GANTT CHART

TABLE 2-1.
CLASSIFICATION OF BATCH PLANTS
BASED ON OPERATING MODE

NUMBER OF PRODUCTS	DECISION-MAKING HORIZON	
	Short Term	**Long Term**
One	Single Product (Custom Order)	Single Product (Campaign)
Many	Scheduled Flowshop Scheduled Jobshop	Multiproduct Multipurpose Multiplant

When a plant produces more than one product in a short term scheduling environment, the operation of the plant must determined by a production scheduler. The products to be produced must be sequenced and assigned to the available equipment units over time. These decisions are made on a batch to batch basis. These plants often produce material to order as well as to inventory. Scheduling objectives might include minimizing the lateness of orders, minimizing sequence dependent cleanouts, minimizing inventory holding costs, and maximizing average throughput. These plants often produce on the order of tens to hundreds of products. The scheduling problem alone for these plants is quite challenging and has been the subject of intense research recently by chemical engineers. Reklaitis (1982) reviews some of the more important scheduling work as it regards batch chemical plants.

In a longer term scheduling environment, batch plants produce material on a campaign basis rather than on a custom order basis. Operating modes are distinguished by the way the equipment is allocated and the number of products being produced at any moment in time. A multiproduct plant produces a number of products in a series of single product campaigns. The processing route through the units in the plant is the same for each product. Thus, a multiproduct plant operates as a single product plant for each product in turn. Figure 2-6 shows a Gantt chart for a three stage, three product multiproduct plant.

Multipurpose plants are more complicated processing systems. A multipurpose plant produces more than one product at a time, and equipment items may process more than one product during any single campaign. In a multipurpose batch plant, the configuration of the process equipment is much more dynamic than in a dedicated, continuous operation. The equipment units can often be reconnected between campaigns. The processing route of a product through the system may also vary depending on the other products being made in the particular campaign. Figure 2-7 shows a Gantt chart for a five unit, three product multipurpose plant.

Multiplant operation occurs when subsets of equipment units may be partitioned into independent plants (single product, multiproduct, or multipurpose) that produce distinct subsets of the entire product slate. Thus, the individual plants are decoupled from each other. The analysis of the overall system can then be broken up into simpler subparts. Campaign #1 in Figure 2-7 is multiplant because two single product plants operate in parallel. Campaign #2 is not multiplant, however, because no subset of equipment units produces a product that is not produced by some other unit.

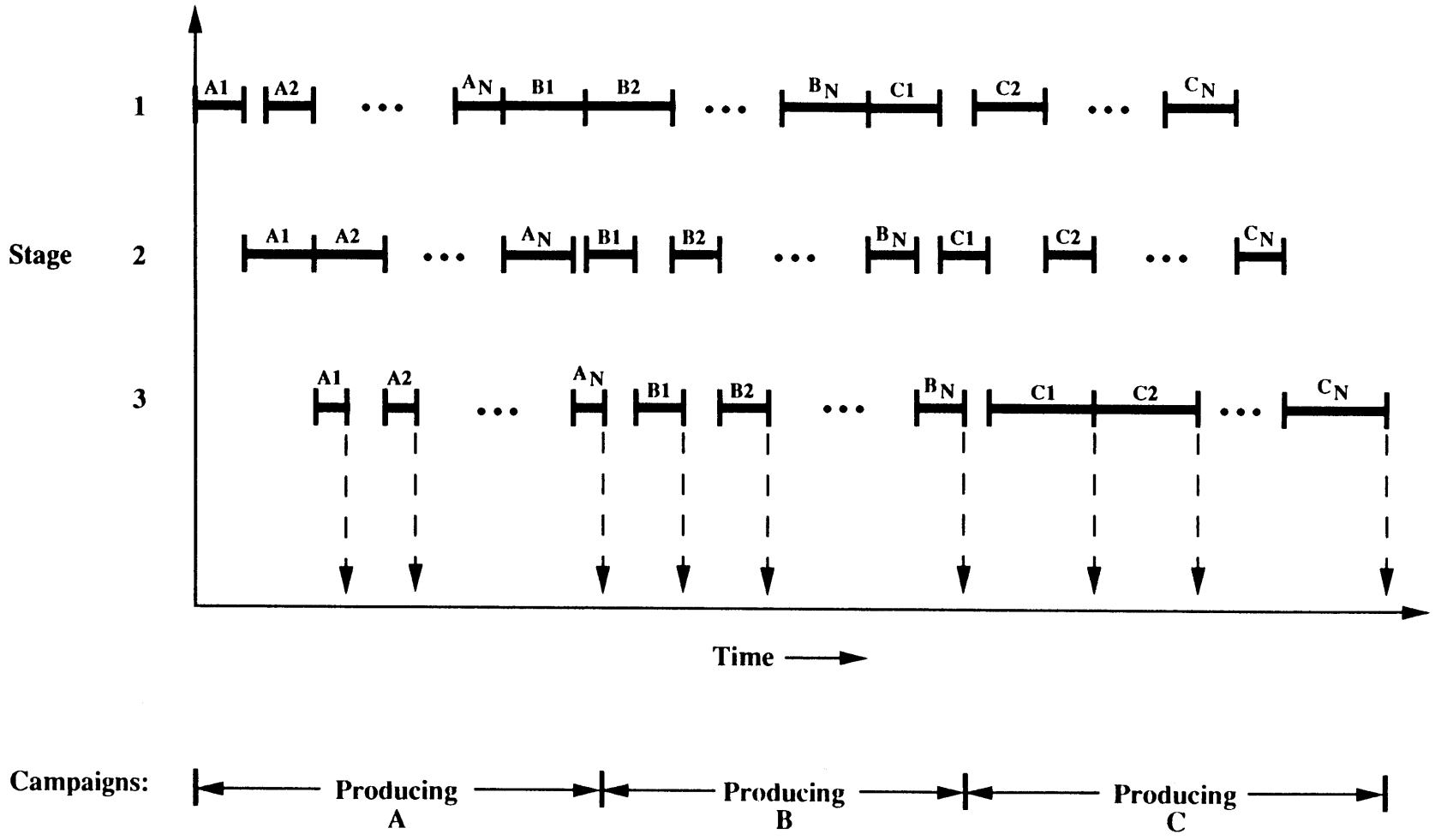


FIGURE 2-6. GANTT CHART FOR MULTIPRODUCT OPERATION

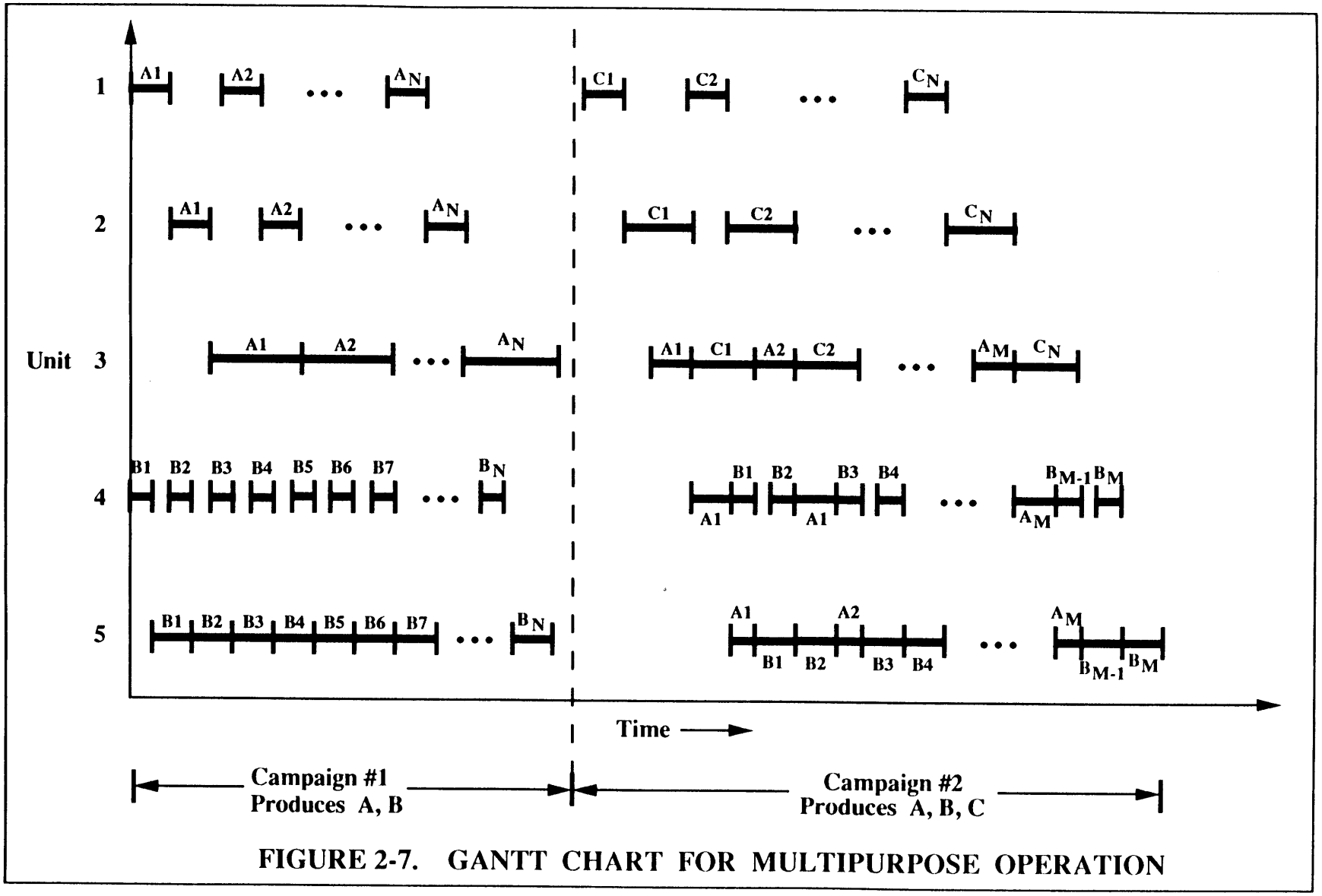


FIGURE 2-7. GANTT CHART FOR MULTIPURPOSE OPERATION

Within a single campaign, there are two types of operation for a non-continuous system. Non-overlapping operation means that the next batch is not started until the current batch has completely passed through the system. Overlapping operation means that the next batch is begun before the current batch clears the system. Thus, there can be a number of batches currently being processed. Overlapping operation utilizes the equipment more efficiently than non-overlapping operation because the effective cycle time is only the longest unit cycle time rather than the sum of all the unit cycle times. Figures 2-8 and 2-9 show Gantt charts that illustrate overlapping and non-overlapping operation for a three stage process producing a campaign of five batches. The overlapping case finishes the campaign 40 percent sooner than the non-overlapping process in this example.

2.1.4 Bottleneck Concept

The bottleneck is a key concept in the analysis of batch processes. Unlike continuous processes that operate at steady state, the various stages in a batch process need not have the same maximum production rate. Idle time can be introduced to slow down stages that are "too fast", or parallel units can be added to speed up "slow" stages. A multitude of options exist to adjust the limiting rate in the process.

The time dependent nature of batch processes and the structural aspects described above affect the way that the average production rate is determined for batch processes. Because batch processes are non-continuous, the production rate must be time averaged. The average rate is essentially the total amount of product accumulated during the campaign divided by the total campaign time. The average production rate can be calculated using information from the product recipe and the process equipment.

The determination of the average production rate is a basic step in the optimal batch process design problem. A discussion of process bottlenecks and the way that they affect the average production rate is given in this section. The following useful terms are defined below: (1) stage cycle time, (2) effective stage cycle time, (3) limiting cycle time, (4) size factor, (5) stage batch size, (6) batch subtrain, (7) limiting batch size, and (8) end effects. These terms are used routinely in the analysis of batch processes.

The ***stage cycle time*** is the time required for a stage to complete a single batch of material, i.e., the time between the starting moments of successive batches. The stage cycle time for stage i (T_i) is the sum of the task time requirements and the cleaning (set up), filling, draining, and idle times for a given stage. For example, suppose two tasks (heating

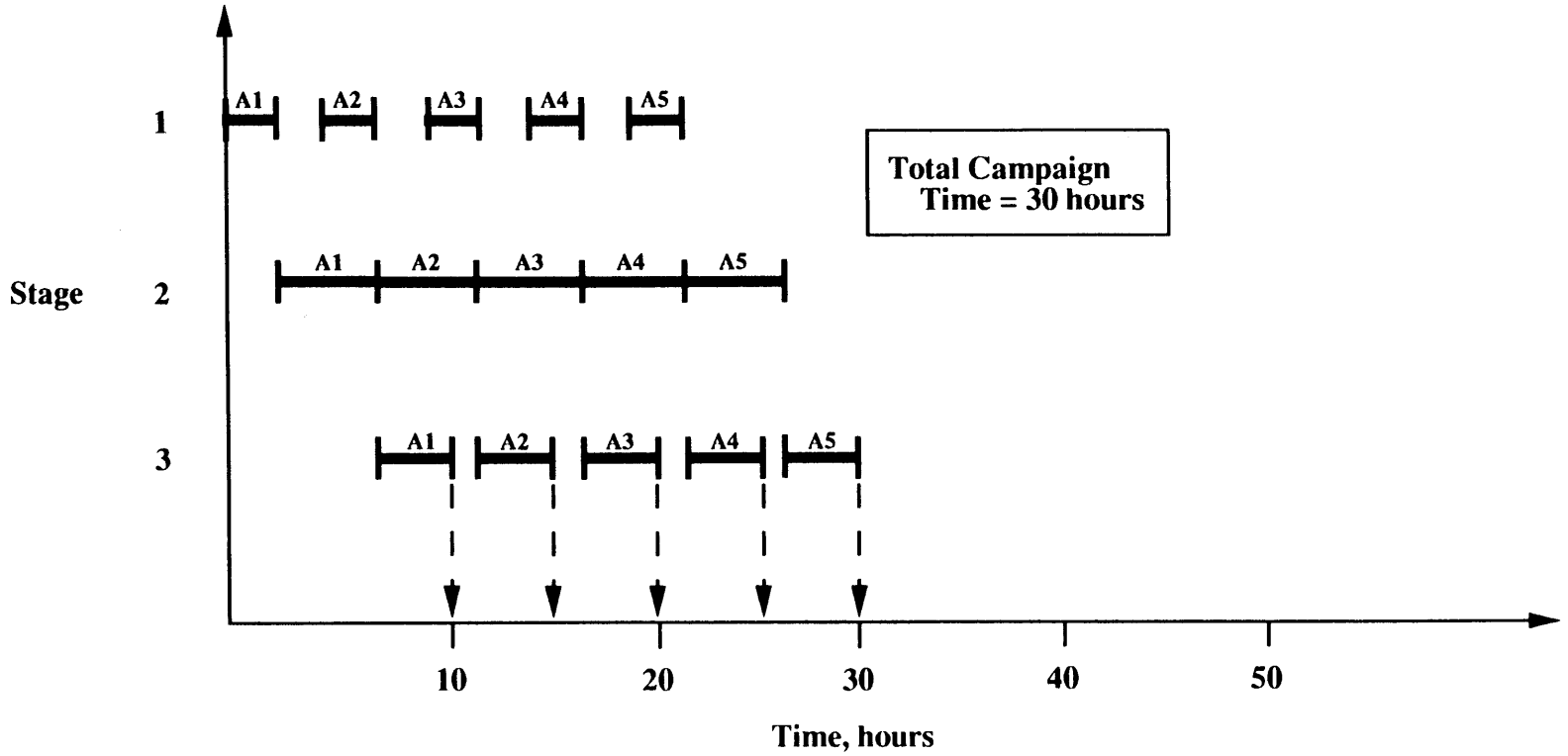


FIGURE 2-8. GANTT CHART FOR OVERLAPPING OPERATION

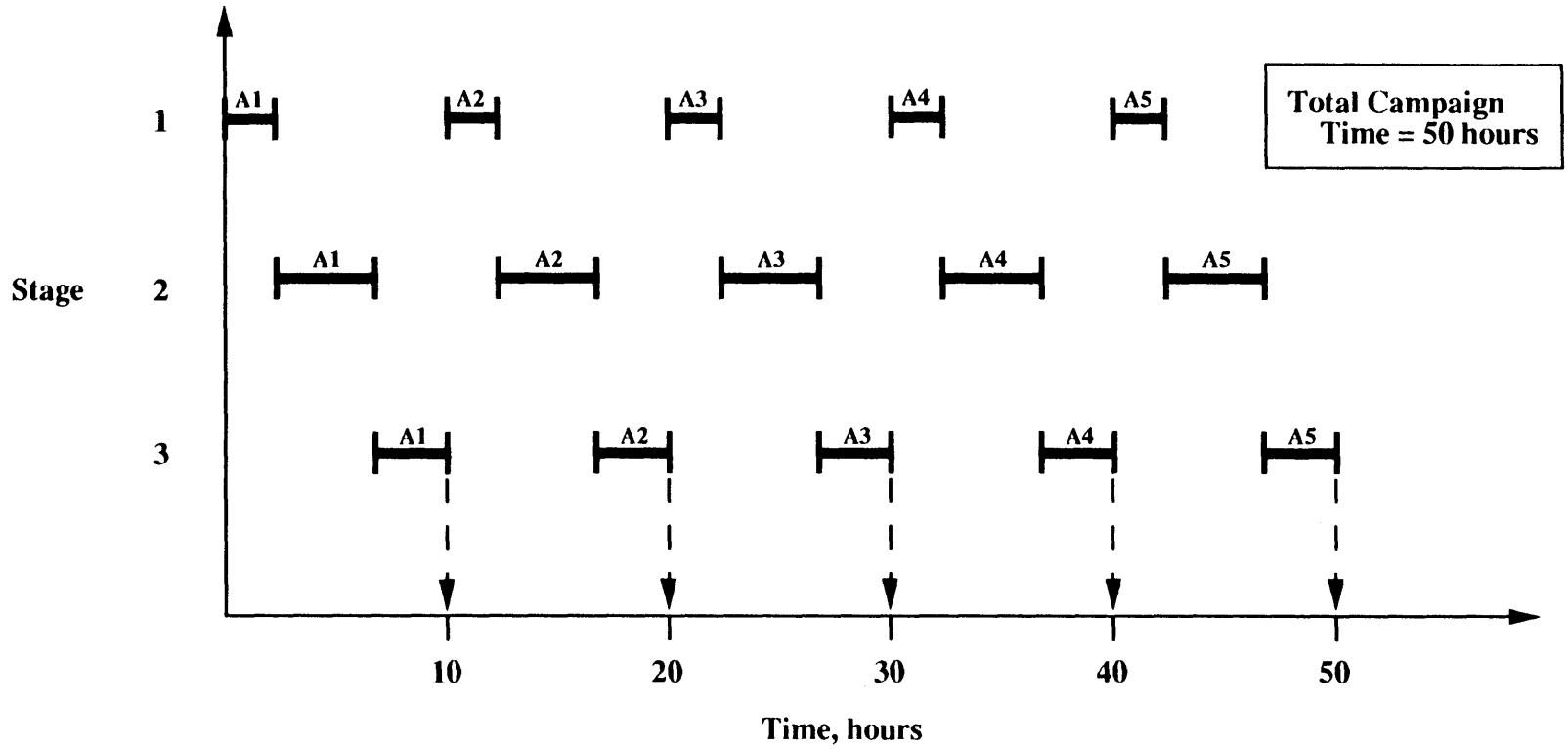


FIGURE 2-9. GANTT CHART FOR NON-OVERLAPPING OPERATION

and reacting) are carried out in a batch unit at stage i. Then, the stage cycle time would be given by:

$$T_i = t_{clean} + t_{fill} + t_{heat} + t_{react} + t_{drain} + t_{idle} \quad (2\text{-}2)$$

where the various times are indicated by the respective subscript. If two units were operating out of phase at stage i, then the ***effective stage cycle time*** ($T_{i\,eff}$) would be half of the value for one unit in parallel. In general,

$$T_{i\,eff} = \frac{T_i}{n_i} \quad (2\text{-}3)$$

for n_i units operating in parallel out of phase at stage i.

For an overlapping non-continuous process with n_S stages and no intermediate storage, one stage will have the longest cycle time. Because there is no intermediate storage, this slow step will limit the entire process. The ***limiting cycle time*** , T_L, satisfies the following relationship:

$$T_L = \max_{i=1,\,n_s} \{T_{i\,eff}\} \quad (2\text{-}4)$$

The non-bottleneck stages have idle time so that all stages operate with the limiting cycle time. This idle time is given by:

$$t_{idle\,i} = T_L - T_{i\,eff} \quad (2\text{-}5)$$

Figure 2-10 shows a Gantt chart for a three stage process with a second unit in parallel operating out of phase at stage 1. The stage cycle times, effective cycle times, and idle times are shown. The limiting cycle time is cut in half by using a parallel unit at stage 1, which is the bottleneck stage.

In practice, there can be stochastic variations in the time requirements for processing tasks. For design work, these times are generally assumed to be deterministic and thus not randomly variable. Units can then be assumed to operate in a perfectly periodic fashion.

The ***size factor*** (S_{ip}) relates the size of the equipment unit at stage i required to produce a unit of product p at the end of the process. The size factor can be determined as

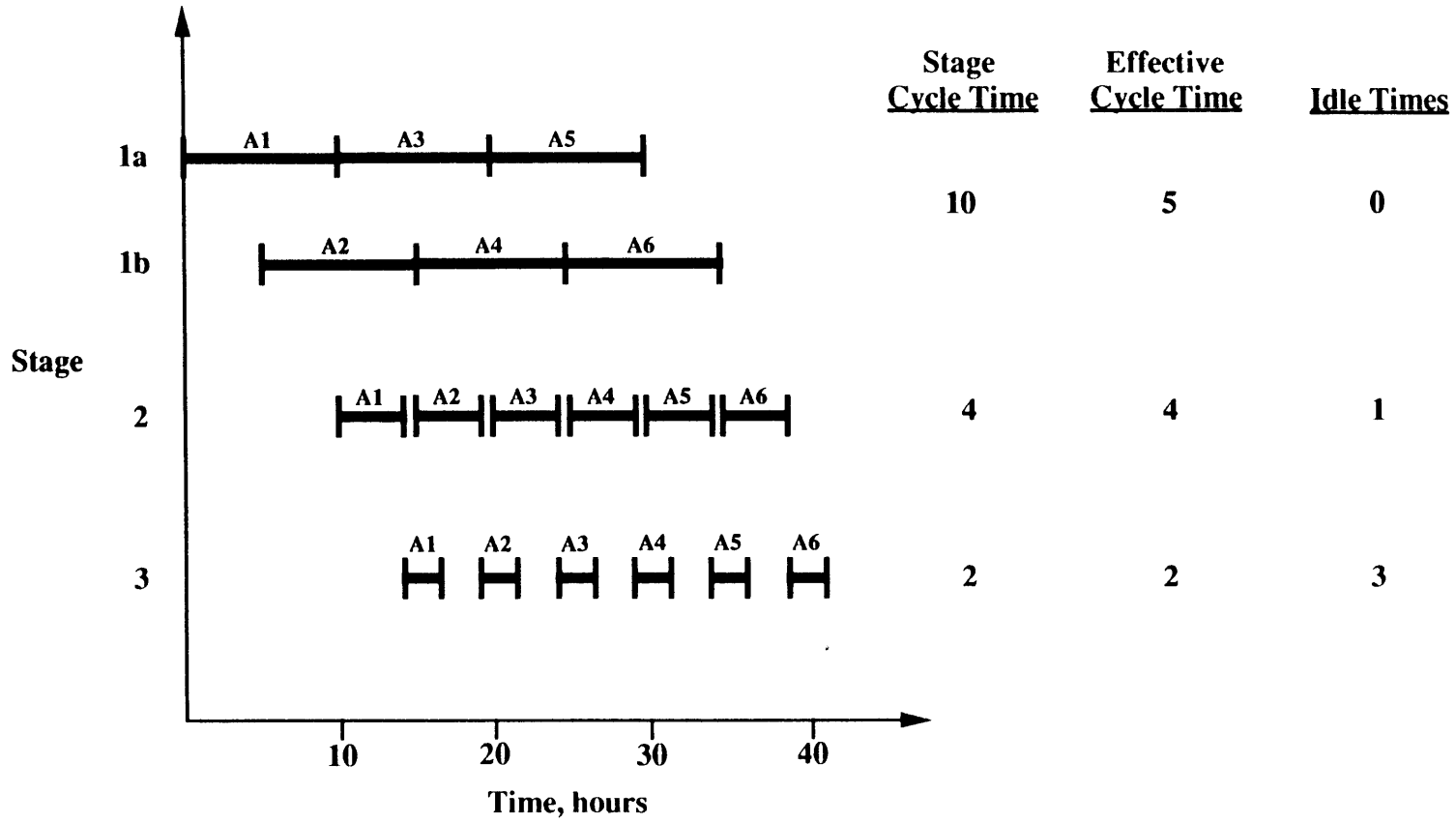


FIGURE 2-10. EFFECT OF ADDING A PARALLEL UNIT OUT OF PHASE

follows. First, the process recipe is followed for some base case amount of reactants. At each stage, the maximum volume of material at any time is noted. The volume of material might vary from stage to stage because of average density changes with temperature or reaction, or because some material is being charged or removed. At the end of the base case run, the maximum volume at each stage is known as well as the amount of product produced at the end of the process. The size factor is a proportionality constant which gives the equipment size at stage i needed to produce a unit amount of final product. The size factor at any stage i is a function of all process operating variables (recipe parameters) and processing times, regardless of the stage they directly affect.

The ***stage batch size*** (B_{ip}) is the amount of final product p that stage i is capable of producing. With no parallel units, the batch size is given by:

$$B_{ip} = \frac{v_i}{S_{ip}} \tag{2-6}$$

where v_i is the volume of material at stage i. The volume at stage i is limited by the size of the unit (V_i) and some lower bound volume (v_i^L) set to insure proper operation:

$$v_i^L \leq v_i \leq V_i \tag{2-7}$$

For example, a batch mixer may require that the volume of material be at least half the volume of the unit in order to have the material properly mixed by the impeller. Generally, units are assumed to run full if possible.

When intermediate storage is placed between adjacent stages, the process is broken into ***batch subtrains***. For example, if M intermediate storage locations are selected, then an n_S stage process will be broken into M+1 batch subtrains. Intermediate storage decouples the limiting cycle times of the subtrains. If the storage is large enough (i.e., UIS), the limiting cycle times of the subtrains will be completely independent.

Equations 2-6 and 2-7 can be combined to generate the following constraint:

$$V_i \geq B_{ip} S_{ip} \tag{2-8}$$

For each batch subtrain or a multistage process with no intermediate storage, each stage must have the same batch size. As the batch size is steadily increased, eventually one of the stages becomes full. Thus, the stage with the lowest batch size limits the entire process.

The ***limiting batch size*** for subtrain m for product p is given by:

$$B_{L\,mp} = \min_{\forall\ i \text{ in subtrain } m} \{B_{ip}\} \tag{2-9}$$

The other stages must therefore run with less than full volumes. Note that when different products are produced on the same equipment, the location of the stage that limits the batch size could vary because the size factors are different for each product.

The average production rate of a batch subtrain is determined as follows for processes without parallel units. First, the limiting cycle time and limiting batch size are determined. These are the bottlenecks in the subtrain. Note that the limiting cycle time need not occur at the stage that limits the batch size. The average production rate for subtrain m for product p is given by

$$R_{mp} = \frac{B_{L\,mp}}{T_{L\,mp}} \tag{2-10}$$

When there is no intermediate storage, the average rate calculated is the rate for the entire process. For cases with intermediate storage, the average rate for the entire process for product p (R_{Lp}) is limited by the limiting rate batch subtrain, as shown:

$$R_{Lp} = \min_{m=1,M+1} \{R_{mp}\} \tag{2-11}$$

These average rates ignore ***end effects***, that is the time to start up and shut down the process.

For the NIS case, the end effects mainly involve the time required to get the first batch through the system. Once the first batch has been produced, succeeding batches are finished every T_L time units. Figure 2-11 illustrates end effects for the NIS case. A three stage process is shown with the given stage time requirements (2, 3, 5 hrs) and a limiting batch size of 100 kg of product. If only a single batch is produced, the actual rate is 10 kg per hr (100 kg divided by 10 hr). As the number of batches increases, the start-up effects diminish and the average rate approaches 20 kg per hr (100 kg divided by 5 hr) as predicted by Equation 2-10 above. In general, end effects can be neglected when a campaign produces a large number (approximately 50 or more) of batches.

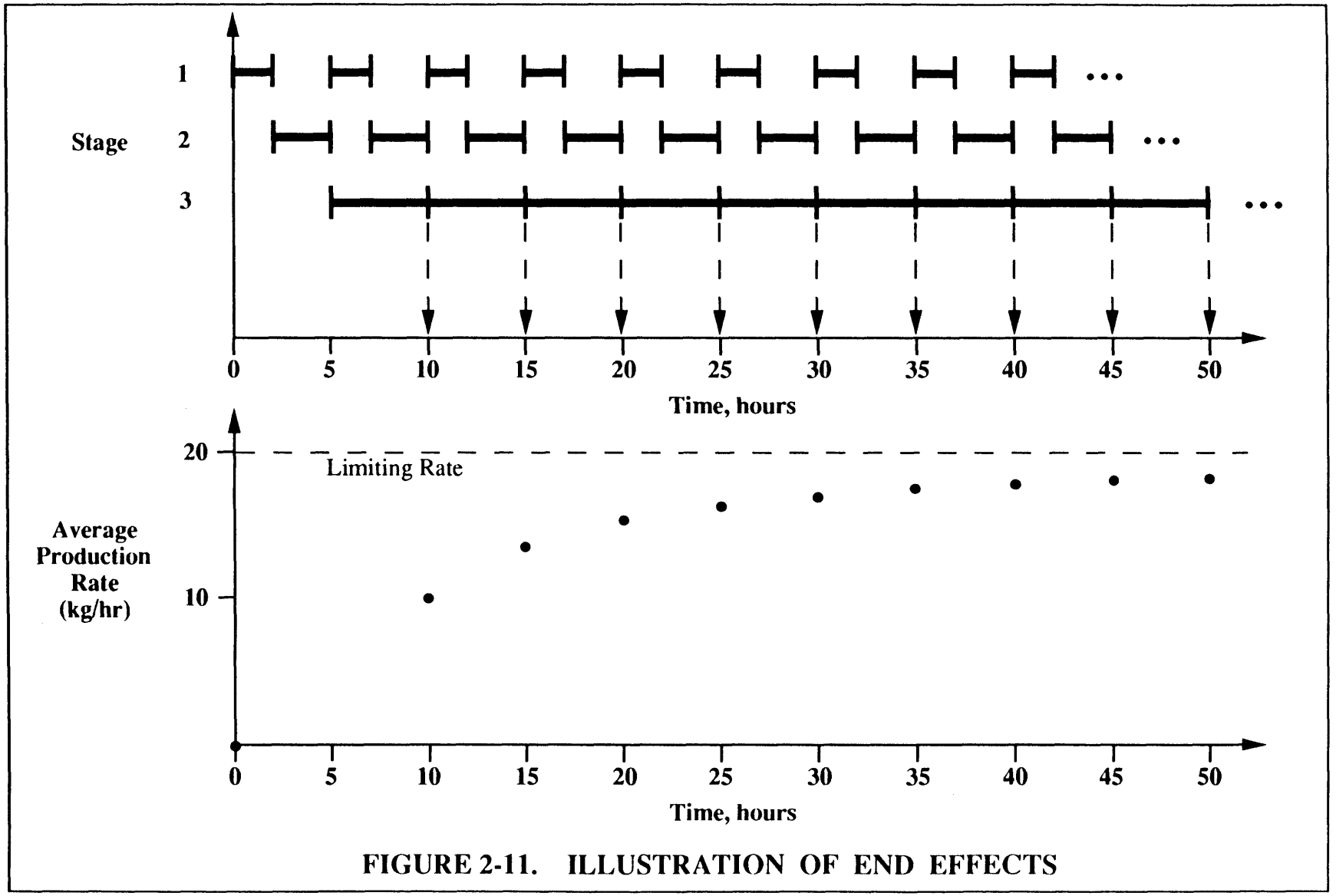


FIGURE 2-11. ILLUSTRATION OF END EFFECTS

The situation is not as clear when intermediate storage is involved. End effects again include start-up and shut-down times. However, these times are harder to quantify because of the presence of intermediate storage. Each subtrain has its own distinct limiting batch size and limiting cycle time. In general though, end effects can safely be neglected when a campaign produces a large number of batches. It is important to realize that developing start-up and shut-down procedures could be a non-trivial task. Less than full batches may be need to be processed to clear storage tanks during shutdown.

In order to minimize the amount of intermediate storage required, periodic operation is generally assumed. The non-limiting rate subtrains are "slowed down" to the limiting rate by either reducing the subtrain batch size (smaller equipment or running partially full units) or increasing the subtrain cycle time (adding idle time). When all the subtrain rates equal the limiting average rate, the minimum required storage volume (V^*_{mp}) between subtrains m and m+1 for product p can be estimated by a simple formula (Takamatsu et al., 1982):

$$V^*_{mp} = S^*_{mp} (B_{L\,mp} + B_{L,m+1,p}) \qquad (2\text{-}12)$$

where S^*_{mp} is the size factor for the stream containing product p leaving the last unit of subtrain m. This estimation is slightly conservative in some cases but fairly close for most situations. Thus, intermediate storage requirements need not be truly infinite to significantly decouple adjacent subtrains. More detailed analyses can be used to get a better estimate of storage costs; these are reviewed later in this chapter.

The concept of the bottleneck is crucial in batch process design and especially optimization. Since the bottleneck stage limits the entire process, the most significant process improvements and cost reductions come from changes made to the bottleneck stage(s). Debottlenecking is an important topic when attempting to improve the operation of an existing batch facility. Likewise, the same basic concepts play an important role in developing good design procedures for batch plants.

2.2 Literature Review

2.2.1 Subtasks in Batch Design

The general batch process design problem is quite difficult. Starting with a recipe for a new product that has been made in the laboratory, the process designer will be faced

with the following questions: (1) how will the new product be made in the plant? (2) how much will it cost? and (3) how fast can it be produced? To answer these questions and come up with a process, the designer must complete a number of steps:

(a) assign recipe tasks to stages of processing
(b) select values for operating variables and times for these tasks
(c) select or size all units and determine the number in parallel at each stage
(d) locate and size intermediate storage (if any)
(e) choose an operating mode for the process

Naturally, these decisions should be made to "optimize" the process, or more generally, the entire plant. Thus, the steps listed above will probably need to be performed a number of times in some type of iterative fashion. In addition, the designer must satisfy constraints imposed on the process and coordinate the requirements of this new product with those of others that will be competing for shared resources in the plant.

There are many possible options and alternatives available among which the designer can choose. Rippin (1983a) described in detail many of the features that complicate the batch process design problem. A number of these fundamental aspects of batch processes have been reviewed in Section 2.1. Unfortunately, no general methodology has been established to handle the entire problem with all its possible complexity. Most research work has focused on smaller, more manageable parts of the overall problem.

Beginning in the 1970's, a growing number of researchers began to address these issues and develop solution approaches for handling various aspects of the general batch process design problem. The literature in this area has been expanding at an accelerated rate during the past decade as more and more workers have entered the field. In their excellent review papers, Rippin (1983b) and Reklaitis (1989) have summarized the major work that has been done on the design of batch processes and batch plants. A brief summary of the major subproblems in the batch process design area is presented here. The goal of this literature review is to provide the context into which this particular work falls.

This thesis focuses on subproblems (b), the selection of values for operating variables and processing times, and (c), the selection or sizing of units and the number in parallel. More specifically, it is the inclusion of the performance issues of subproblem (b)

with the better studied structural concerns of subproblem (c) that will be highlighted. For this reason, more attention will be placed on previous work in these two sub-areas.

2.2.2 Optimal Sizing Problem

The first subproblem to be reviewed is subproblem (c), which shall be referred to as the "optimal sizing" problem. This subproblem was the first area to be addressed in the literature. Pioneering papers on the optimal design of batch processes were published by Ketner (1960), Loonkar and Robinson (1970), Robinson and Loonkar (1972), Sparrow et al. (1975), Grossmann and Sargent (1979), and Knopf et al. (1982). With exceptions to be noted later, the basic problem addressed by all of these investigators is the design of a new grass roots multiproduct plant with the following input:

(1) A slate of P products with specified quantities of each is to be produced over a time horizon H. Each product is made in a single product campaign.
(2) The recipes for the products are expressed as a process flowsheet with specified types of equipment in a given sequence.
(3) All process operating conditions are set, and performance constraints are implicitly satisfied.
(4) The processing times in each unit for each product are specified either as a constant or as a function of the amount of material processed in the batch.
(5) The plant operates with no intermediate storage (NIS).
(6) The cost of each type of equipment is given as a function of its size (batch) or processing rate (semi-continuous).

The goal of the optimization is the determination of the size of each piece of equipment so as to produce the required slate of products in a plant with minimum capital investment.

The first work on the optimal sizing of batch chemical processing systems was done at American Cyanamid by Ketner and Loonkar and Robinson. Their formulation of the optimal sizing problem became the basis for virtually all subsequent work in this area. Understanding their initial contributions will make clear the value of later improvements.

Ketner (1960) addressed the minimum capital cost problem for a single product plant consisting of batch and semi-continuous units. The batch operating times were fixed, and the process was assumed to operate in a non-overlapping mode. The capital cost per unit rate of production was minimized using an iterative procedure. The operating times of the semi-continuous units were "optimized" on each iteration for linearized capital cost

functions. All units were then sized by applying the production rate constraint, and new linear cost functions were determined. The optimal solution was obtained in an average of three to four iterations according to the author. The optimization procedure was based on the minimization of the objective function using calculus.

Loonkar and Robinson (1970) improved on Ketner's work by introducing power law cost functions. They also attacked the minimum capital cost problem for a single product plant consisting of batch and semi-continuous units. Although they presented their analysis for non-overlapping operation, they discussed the modifications required to handle overlapping operation as well. Loonkar and Robinson were also the first to include the concept of size factors. Their analysis assumed that the size factors were given, an assumption equivalent to fixing the process performance. They optimized the sizes of the batch units and the rates of the semi-continuous units for a specified production rate using principles from multi-variable calculus.

Hellinckx and Rijckaert (1971) were the first to apply mathematical programming techniques to the optimal batch process design problem. They showed in a short note that the single product optimization problem posed by Loonkar and Robinson could be formulated as a geometric programming problem. By posing the problem as a geometric program, the solution to the optimization problem could be proven to be the global optimum. To demonstrate the utility of geometric programming algorithms, the authors re-solved the example problem from Loonkar and Robinson.

Robinson and Loonkar (1972) extended their previous work to handle multiproduct plants. The optimal design was constrained to satisfy the annual production requirements for each product. In addition, differences in the performance of the units for the various products were accounted for by size factors for each unit for each product. The sizes of the process units were determined so that they could handle all the products. Thus, some units would be oversized for parts of the product slate. The authors used a multivariable search procedure that could handle inequality constraints, in particular a modified version of a technique developed by Hooke and Jeeves (1961) and an accelerated gradient procedure developed by Newberger (1969).

Rippin's group at ETH Zurich was the first to build on the work of Loonkar and Robinson. Sparrow, Forder, and Rippin (1975) addressed the minimum capital cost problem for cases when equipment units are available only in discrete sizes. The authors pointed out that many batch equipment units are standard items that are not custom

designed. Instead, they are purchased from vendor catalogues and are only available in certain specified sizes. In this case, all the decision variables in the capital cost minimization problem are discrete, generating a combinatorial optimization problem.

The key optimization variables were the number of parallel units at each stage and the sizes of the batch units. However, parallel units were constrained to be identical in their work. This restriction greatly reduced the number of possible structures. The batch operation times were given for each product, and the semi-continuous units were sized by assuming that their operation times were given. Process performance was fixed because the unit operation times and the size factors were all given.

Sparrow et al. (1975) proposed two solution methods for this combinatorial optimization problem. First, they developed a branch and bound procedure which guaranteed that an optimum was found. Branch and bound methods can require excessive amounts of computational effort for problems with large numbers of combinations. If good bounds can be determined, the pruning of the search tree can be sped up considerably. Sparrow et al. used a heuristic method to generate a good initial solution and also improved the efficiency of the search algorithm by branching in certain ways.

The heuristic solution procedure that they developed represented a second solution method on its own. Near-optimal solutions (within a few percent of the optimal solution) were obtained in much less computation time. The heart of the heuristic method was the use of size factors for a single hypothetical product which represented the average characteristics of the multiple products to be produced.

Grossmann and Sargent (1979) formulated the minimum capital cost problem for a multiproduct plant as a mixed integer nonlinear programming problem (MINLP). They showed that the relaxed NLP problem could be posed as a geometric programming problem. The volumes of the batch units, the product batch sizes, and the number of parallel units at each stage were the optimization variables. Semi-continuous units were not included. However, the batch unit operation times were allowed to be functions of the batch size.

Grossmann and Sargent solved a series of relaxed subproblems to obtain integral solutions for the number of parallel units. First, the number of units at each stage was allowed to vary continuously, and the resulting "continuous" subproblem was solved using a nonlinear programming (NLP) technique. The authors then coupled a branch and bound

procedure with the NLP solution method to obtain an optimal solution with an integral number of parallel units. Good suboptimal solutions were often obtained quickly, and thus the need to do an exhaustive branch and bound search could be avoided in many cases.

Knopf, Okos, and Reklaitis (1982) at Purdue improved upon the solution procedure developed by Grossmann and Sargent by including semi-continuous units in their analysis. Knopf et al. also formulated the minimum capital cost problem for a multiproduct plant as a geometric programming problem. The usual fixed performance assumptions were made, including fixed batch times and given size factors. They solved the convex primal formulation after making posynomial substitutions to generate linear constraints. A generalized reduced gradient optimization procedure was used to solve the problem.

Knopf et al. (1982) also included operating costs in the objective function for the first time. They looked at a cottage cheese production facility and found that energy costs dominated the optimal solution. The solution was very sensitive to the form of the empirical model for the energy costs, with the number of parallel units for the key stage varying from three to eight. This work demonstrated that better process modeling is required to handle operating costs and that operating costs should not always be neglected in the optimal sizing problem.

A number of other workers have addressed aspects of the optimal sizing problem. Flatz (1980, 1981) described some short-cut hand calculation procedures for estimating the equipment sizes for a multiproduct plant. Wiede et al. (1981) compared various heuristic and branch and bound strategies for dealing with the combinatoric aspects introduced by parallel units and discrete unit sizes. A better short-cut method for obtaining near-optimal solutions for the sizes and number of parallel units for single product plants was developed by Yeh and Reklaitis (1987). Espuna et al. (1989) and Modi and Karimi (1989) also developed short-cut solution procedures that obtain good solutions to the optimal sizing problem for multiproduct plants.

Recent extensions of the optimal sizing problem have addressed the problem of design under uncertainty. In particular, the uncertainty or seasonability of anticipated demands for the products to be produced by the multiproduct plant have received special attention. Vaselenak (1985), Reinhart and Rippin (1986,1987), and H. Wellons and Reklaitis (1989) have all worked on this particular problem. Staged expansion of the plant by adding equipment in parallel was usually considered in order to satisfy product demands

that were higher than expected. Potentially larger revenues from the sale of products had to be weighed against increased capital investment costs.

2.2.3 Optimal Sizing with More General Operating Modes

After the initial work described above on multiproduct plants, later workers considered the design of batch plants with more complicated operating modes. These new aspects introduced scheduling considerations into the batch process design problem. The first extensions dealt with multipurpose plants. Later work has begun to focus on shorter term scheduling considerations in the design of multipurpose plants. The work on designing batch plants with operating modes that require scheduling considerations is reviewed in this section.

For multipurpose plants, products continue to be made in long term campaigns, but multiple products are made during single campaigns. This fact introduces the need to solve the campaign formation problem, i.e., determine the groups of products that should be produced together in single campaigns. In addition, equipment units in multipurpose plants are often reconfigured between campaigns. Thus, for each campaign, the equipment units must be allocated to stages in the processing of the various products.

A number of workers (Mauderli and Rippin, 1979, 1980; Rich and Prokopakis, 1986, 1987; Lazaro and Puigjaner, 1985; Suhami and Mah, 1984; and M. Wellons and Reklaitis, 1989a,b) have worked on aspects of either the campaign formation or the equipment allocation problem. These workers focused on the scheduling aspects of the multipurpose plant and did not size equipment units. M. Wellons and Reklaitis (1989a,b) noted that the use of non-identical parallel units with no intermediate storage could create situations where the limiting batch size would be path dependent. In addition, the maximum production rate was not always obtained by running units full. In fact, determining the optimal operating batch sizes for these situations was not a trivial exercise even though the equipment sizes were given.

A more difficult challenge is the design of a multipurpose plant. In this case, the sizing of the equipment must be addressed in addition to the campaign formation and equipment allocation problems. A number of the more significant contributions are reviewed here.

Suhami and Mah (1982) looked at minimizing the capital cost of a multipurpose plant with only batch units. Operation times were fixed, and units were available in a

continuous range of sizes. Suhami and Mah formulated their problem as an MINLP problem. A heuristic procedure generated good product configurations for the various campaigns. Campaigns with more than one product were considered feasible only if the products did not require any of the same equipment types. A generalized reduced gradient method was applied to the continuous relaxation of the problem. The discrete variables were then forced to integral values by adding extra constraints. The method obtained near-optimal solutions for relatively small problems. However, a time-consuming branch and bound solution procedure was avoided.

A number of workers have attempted to either improve on the solution procedure proposed by Suhami and Mah (1982) or broaden the scope of the problem by relaxing restrictive assumptions. Imai and Nishida (1984) proposed a set partitioning method to determine the configuration of the best campaign structure. Klossner and Rippin (1984) enumerated all possible campaigns by solving a set partitioning problem. For each configuration, a MINLP was formulated and solved. Vaselenak et al. (1987b) used a superstructure approach to form all possible product configurations. Limiting horizon constraints are formulated and used in an MINLP formulation to optimally size the equipment units. Faqir and Karimi (1989a) developed a method based on the theory of linear inequalities to derive the complete set of limiting horizon constraints. This method avoided the possible screening out of some configurations in cases where the method of Vaselenak et al. did not work. Coulman (1989) also proposed an improvement to Vaselenak's method for generating the horizon constraint set.

Recent workers (Janicke, 1987; Kiraly et al., 1988; Faqir and Karimi, 1989b; Cerda et al., 1989; and Papageorgaki and Reklaitis, 1989) have addressed cases with more general equipment allocation schemes for the design of multipurpose plants. They have built on the formulations and methods described above. Performance was still fixed in the form of given size factors and processing time relationships (function of batch size only). Solution methods have generally involved the use of mathematical programming methods. Both Faqir and Karimi (1989b) and Papageorgaki and Reklaitis (1989) considered the case when equipment units are only available in discrete sizes.

Comparatively much less work has been done on the design of batch plants that operate with scheduling decisions made on the batch to batch time scale rather than on the campaign time scale. However, previous workers have demonstrated improvements in operating efficiencies by making equipment and scheduling modifications to batch plants. Typically, discrete event simulators have been used to examine alternative process

structures, operating policies, or scheduling rules (Fruit et al., 1974; Embury et al., 1976,1977; Overturf et al., 1978a,b; Felder, 1983a,b; and Kuriyan et al., 1986). Most of these simulation studies have also incorporated stochastic variations in process behavior. These papers indicated that design and scheduling decisions can have significant interactions that affect process performance.

These interactions were addressed by Knopf, Reklaitis, and Okos (1984) when they examined a fluids processing system which produced various milk and juice products. Intermediate storage as well as the order in which the products were processed played an important role in the operation of the plant. The objective was the minimization of the capital cost for the plant, including the costs of the storage units. The authors simulated the operation of the system using SLAM, a higher level simulation language. The discrete event simulator basically kept track of the amounts of material in each unit and used dispatching rules to route batches from one stage to the next. Because the plant was regenerative (i.e., short operation cycle), a simulation method could be used without requiring excessive computing time. A quadratic surface response strategy was used to generate improved solutions.

Birewar and Grossmann (1989a) considered the use of mixed product campaigns in order to reduce the capital cost of a multiproduct plant. The plant was restricted to single units at any stage, and a large number of batches of each product was assumed. Normal multiproduct operation assumes that products are made in a series of single product campaigns. Typically the plant must finish all the campaigns within some given horizon time. Because the units are sized to minimize the capital cost, the plant will generally use the entire time available. By considering ways of scheduling the products through the plant a batch at a time, a sequence could be found that would reduce the total production time needed to produce all the products. The equipment units could then be made smaller and reduce the capital cost. Birewar and Grossmann generate new constraints that incorporate the effects of doing this scheduling into an NLP formulation of the minimum capital cost problem.

2.2.4 Intermediate Storage Considerations

Intermediate storage plays an important role in batch processes as discussed in earlier in Section 2.1.2. The use of storage can have a significant impact on the sizing of the process equipment because of the way storage affects the average production rate. Proper use of intermediate storage can reduce unit idle times and the sizes necessary for

units to satisfy the average production rate constraint. However, this storage capacity is not free. There is a trade-off that must be explored, i.e., the size reductions gained versus the additional cost of storage. In order to examine this trade-off, methods for sizing storage are required. This section reviews work that has addressed the sizing of intermediate storage or the incorporation of storage effects into the optimal design problem.

Takamatsu et al. (1979) considered the case of a continuous process with a single batch stage of N parallel units. Storage was placed before and after the batch stage. The problems of determining the proper timing of the parallel batch units and the minimum required storage volumes were addressed. For identical parallel units, analytical solutions were obtained. However, solving the problem for non-identical units involved a very large direct search. The authors reduced the size of this search by applying a theorem they developed.

Karimi and Reklaitis (1983) developed a procedure for determining the minimum size intermediate storage tank. Their method involved the use of Fourier series representations for the hold-up in the tank at any time. The sizing was done deterministically for given feed and discharge rates for the tank and given cycle times for the upstream and downstream units. The problem is quite complicated because the minimum storage tank volume is a discontinuous function. The authors concentrated solely on the intermediate storage tank sizing problem and did not attempt to incorporate this sizing method into the more general batch design problem. Takamatsu et al. (1984) and Karimi and Reklaitis (1985a,b,c) have continued their work on sizing intermediate storage, extending their investigations to include stochastic variations and the effects of process upsets on storage tank levels.

Takamatsu, Hashimoto, and Hasebe (1982) were the first to include the sizing of storage tanks in the optimal sizing problem. They attempted to minimize the capital cost of a single product plant. The batch times were assumed to be a function of the size of the batch units. Semi-continuous units were not sized, and the processing rates of pumps feeding and draining the storage tanks were also assumed known. The optimization variables were the number of parallel batch units at each stage, the sizes of the batch units, and the sizes of the storage tanks.

The inclusion of intermediate storage significantly complicated the problem because the initial tank volumes and the relative starting times of the upstream and downstream unit operation cycles became important variables that directly affected the tank sizes. Because

the minimum storage size is in general a discontinuous function, the solution procedure proposed consisted of an "intelligent" direct search. The authors reduced the search space through their analysis before starting the direct search procedure.

As mentioned in the previous section, Knopf et al. (1984) also included the sizing of intermediate storage in the design of the fluids processing plant. The storage size and the unit sizes were obtained by minimizing a quadratic surface response model. The surface response model estimated the effect of the storage and process sizes on process operations. Feasibility was insured by running the SLAM simulation model of the process for the "optimized" unit and storage sizes.

Yeh and Reklaitis (1987) considered the use of intermediate storage to increase the average production rate (and consequently reduce the capital cost) of single product batch/semi-continuous plants. Batch subtrains were required to operate at the same average production rate in order to keep storage sizes reasonably small. The cost of intermediate storage was also assumed to be negligible in comparison to the costs of processing units, although the authors recognized that the addition of storage should be limited to the locations where it would result in the greatest cost savings in terms of size reductions for process units. They focused on the optimal location of intermediate storage, adding storage in evolutionary fashion until diminishing returns were reached. Heuristic rules were used to guide the evolutionary method, and short-cut models were used to estimate cost reductions.

Modi and Karimi (1989) used sizing expressions that gave slightly conservative estimates of the intermediate storage volume in an MINLP formulation of the optimal sizing problem for multiproduct plants. The objective function was the minimization of the capital cost for the processing units and the intermediate storage. The locations of the storage units were assumed to be given. They also developed a heuristic procedure that obtained near-optimal solutions without the need to solve a mathematical programming problem. This heuristic procedure was based on their short-cut sizing method and used a series of single variable line searches to optimize the relative production rates of the various single product campaigns.

2.2.5 Task to Stage Assignment

The task to stage assignment problem was described in Section 2.1.2. By merging or splitting adjacent tasks, the limiting cycle time for a batch subtrain can be modified. This

change in the limiting cycle time consequently affects the size of the equipment needed to produce the product requirement within the campaign time horizon. Also, merging and splitting tasks affects the total number of units required by the process. The proper task to stage assignments must be made to generate a minimum capital cost process.

Yeh and Reklaitis (1987) were the first to include the task to stage assignment problem together with the optimal sizing problem. However, they noted that Bamopoulos et al. (1981) were actually the first to address the issue. Bamopoulos et al. illustrated various task to stage assignments in the course of a trial and error, case study approach. Yeh and Reklaitis developed systematic procedures that incorporated task merging and splitting into the minimization of the capital cost for single product plants. The combined task assignment and equipment sizing problem can be formulated as an MINLP and solved using mathematical programming methods. Yeh and Reklaitis also developed a heuristic technique that obtained good, near-optimal solutions through the use of heuristic rules and short-cut models.

A few recent workers have incorporated the task to assignment problem into their methods for designing more complicated processing systems. Papageorgaki and Reklaitis (1989) incorporated task merging and splitting in their mathematical programming approach for the design of multipurpose plants. Birewar and Grossmann (1989b) also included the task to stage assignment problem in their mathematical programming approach for the design of multiproduct plants with mixed product campaigns. Both of these recent efforts developed MINLP formulations that simultaneously addressed three of the five subproblem areas mentioned in Section 2.2.1.

2.2.6 Choosing Operating Characteristics

The main focus of this project is the coupling of the determination of values for process operating variables and times with the selection of sizes for the equipment units in an overall optimization scheme. These operating variables and processing times affect the performance of the process. Very little work in the literature has even addressed the problem of including performance issues in the optimal design of batch processes. Although he did not incorporate performance optimization into a design scheme, Rippin (1983a,b) discussed the modeling of batch systems to determine their performance. This section will review Rippin's comments on performance and the papers that include process performance issues.

Rippin took a broad look at batch chemical processes in his review article (1983b). He discussed topics that ranged from the very detailed (modeling the operation of a single batch unit) to the very general (design of multipurpose plants). The topic of unit performance came up at the more detailed end of this problem spectrum. Rippin reviewed previous work that developed methods for optimizing the performance of individual units. The optimization decision variables in these analyses generally included operating variables, such as reaction temperature, reflux ratio, and reactant addition rate, and the processing time. These problems often involved the determination of an optimal operating profile. Possible objectives might include maximizing the unit performance with a given cycle time or minimizing the cycle time required to attain some specified performance.

Rippin (1983b) cautioned that "a cycle time optimization for an individual unit should not be undertaken without considering its potential effects on the performance of other units which operate concurrently." In his article on problem structure, Rippin (1983a) again cautioned against ignoring the global effects of changes in a single unit's performance. He also pointed out that there might be an "opportunity to trade off a better performance against a longer time requirement" for a given unit.

Since the time requirement for a processing stage is such a critical operating variable, it follows that process performance should be optimized in order to produce an optimal design. Rippin (1983a) stated that the performance of a unit could be a function of the unit's operating time, the unit's size, and the properties of the inlet stream (i.e., the performance of upstream units). To account for all these factors, detailed models for the operation of each unit would be required. Optimizing the performance of the entire process and optimally sizing all the units would be quite difficult in Rippin's opinion:

> Such models could, in theory also be used in a larger scale co-ordination of different equipment items and different products ... , but the computational requirements would soon become unmanageable and furthermore any qualitative feel for important aspects of the problem would quickly be submerged in model complexity.

Rippin (1983a) proposed a hierarchical approach to solving these complicated problems. More detailed models would be used for optimizing individual units while more general models would be used for the larger problem. However, no general method was put forth that would allow for the systematic optimization of the total process performance and the equipment sizes. The trade-offs that Rippin had mentioned previously were not really considered.

Sparrow, Rippin, and Forder (1974) developed a computer program that performed sizing and costing calculations for multiproduct batch plants. MULTIBATCH also carried out material and energy balances for the process units to determine the process performance for the given user inputs. The process models took the form of split fractions or user-provided subroutines. The heat and mass balances generated the size factors for the stages and the time requirements for the batch units to process a specified quantity of product.

Sparrow et al. (1974) have included the effects of performance with this program, although with a number of shortcomings. The program calculates size factors (i.e., the system performance characteristics) for a reference amount of material and assumes that the values obtained are independent of scale. More significantly, the program does not consider performance and sizing together. The program operates in a straight-through fashion, first determining the performance for given user inputs and then determining the equipment sizes. No performance optimization is carried out by the program. One last limitation of the program is that it does not include the rates of the semi-continuous units in the optimization. These units are sized based on some user-specified average operating time. Despite these limitations, MULTIBATCH was the first computer package developed for application to the batch process design problem.

Rippin (1983b) illustrated the concept of extending performance and cycle time optimization to a sequence of units operating with no intermediate storage. Although each unit has its own "optimal" operation time when considered alone, the limiting cycle time for a process is given by the longest of these unit operation times. All non-bottleneck units then have idle time. Rippin showed that in general the optimal limiting cycle time is not given by any of the "optimal" individual unit operation times. Instead, the overall system performance must be optimized to give the best limiting cycle time.

Hatipoglu and Rippin (1984) provided more details on this approach. Optimal profiles were incorporated into their method, and both non-overlapping and overlapping cases with no intermediate storage were considered. Attainable regions in the space of performance attributes were constructed by using spline fits of the results of optimal profile determinations. Objective functions that satisfied certain general requirements were maximized by determining the best cycle time. This paper was the first to explicitly consider interstage performance effects on the choice of limiting cycle time.

There were some shortcomings to this paper. First, it appeared that the performances of downstream units were independent of the performance outputs from upstream units. Since this situation is not generally observed, it seems that the determination of the composite attainable region for the entire process would be more difficult to construct. Second, profit-based objectives may not always satisfy the requirements specified by their method. Finally, no mention is made of equipment sizing or the effects that the equipment sizes could have on process operations.

Linnhoff et al. (1987) gave a general discussion about the benefits of debottlenecking batch plants. Debottlenecking an existing plant could be brought about by making structural changes, scheduling modifications, or changes to processing conditions. Linnhoff's comments seemed to advocate a general way of thinking about how one would approach a debottlenecking project rather than a detailed strategy of how to proceed. Very little detail was provided on the actual methods used for making process improvements once the bottlenecks were identified. However, two successful case studies were cited (Clayton, 1987a,b) and briefly described.

Reklaitis (1989) has also noted the potential benefits of making changes to process operating conditions in his recent review paper. The inclusion of these operating variables would probably be most easily done when retrofitting an existing plant according to Reklaitis. Young and Reklaitis (1989) demonstrated the application of BATCHES, a general batch process simulator, to an industrial retrofitting project. Significant improvements in production capacity were achieved, due in part to changes in operating conditions and shorter processing times. These improvements were obtained using a case study approach with the use of the simulator.

Wilson (1987) was the first to include detailed performance modeling for a batch process in an attempt to demonstrate economic trade-offs during design. Implicit in all previous design work were the restrictions that each unit operated with given processing conditions and processed a batch to some fixed endpoint specified by recipe. Wilson included process operating variables and the reaction time as main variables for optimization in his study of a one stage process.

The process was a complex reaction with distillation in a single semi-batch unit for the production of a pharmaceutical intermediate compound. A detailed process model (including kinetics and distillation) was required to determine the effects of varying

operating variables (such as the time, reaction temperature, and initial reflux ratio) on the cost. A number of trade-offs were clearly demonstrated. Wilson used a case study approach to select the best values for process operating variables. No systematic optimization method was put forth. Also, because the process contained only one stage, many of the complicating interactions caused by multiple stages that characterize batch processes were avoided.

2.3 Chapter Summary

This chapter has reviewed the main aspects of batch processes that distinguish them from continuous ones. Basic terms have been defined to provide necessary background information. Batch process design has been broken down into five subtasks that must be carried out. The batch process design literature has been divided into analogous categories. For each one, the work that has advanced better methods for solving that particular aspect of the overall batch process design problem has been reviewed.

The optimal sizing problem has been well studied for both multiproduct and multipurpose plants. Recent work has focused on incorporating more general operating modes (especially shorter term scheduling) in the design of multipurpose plants and on performing task to stage assignments in conjunction with equipment sizing. However, virtually all of this work has assumed that size factors are known and given. Thus, the interactions that exist between process performance, as given by the selection of process operating variables and processing times, and the sizing of the equipment units, have not been properly addressed. Only Wilson (1987) has considered the economic effects that the operating variables have at the design stage.

Chapter 3

PROBLEM FORMULATION

3.1 Batch Production Environment

3.1.1 Dynamic Nature

The production environment of a multipurpose batch plant is a dynamic one. One major advantage of batch plants over continuous ones is increased flexibility. A large number of products can often be made from a set of common, standardized units. Uncertain or changing product demand levels can be handled by adjusting production levels of the various products in the product slate. As the demands for older products decrease, new products or improved product grades can be introduced to the product slate. Eventually, the older products are no longer produced.

The configuration of the process equipment in a multipurpose batch plant is also much less rigid than in a dedicated, continuous operation. The equipment units can often be reconnected when switching from one production campaign to another. Standardized units may also be able to perform a variety of processing tasks. There are then many possible ways to allocate the equipment units to the processing stages for the various products.

As discussed in Section 2.1.3, there are a number of possible operating modes for batch plants. Some plants operate to produce custom orders. In these facilities, orders for products are scheduled on the equipment by a plant scheduler. Other plants produce material on a campaign basis rather than on a custom order basis. Plants running long campaigns are easier to operate and design than plants producing batches to order because detailed production scheduling is not required. Plants operating with this longer term scheduling horizon (campaigns) are assumed for this thesis.

3.1.2 Stagewise Plant Expansion

Although it may happen occasionally, we believe it unlikely that companies will design and build new grass roots multipurpose batch plants to produce a slate of P products when P is a large number. The more likely scenario is that plants will evolve over time, with increased production capabilities added as warranted by increased profitability or by significant additions to the product slate. In this case, the purchase of new equipment to increase plant production capacity would only occur at a series of discrete points in time.

This stage-wise expansion, illustrated in Figure 3-1, would involve a series of debottlenecking and retrofit design projects. After new equipment units are purchased, the plant would be under-utilized for a time. Equipment utilization would rise over time as production levels of existing products increased or new products were added to the product slate. Eventually, plant production would again be limited by the capacity of the equipment, which would now be nearly fully utilized. Debottlenecking studies might commence in order to squeeze more production out of the existing facility. Finally, when no additional debottlenecking or better scheduling could free up enough time for further desired production increases, the plant would be expanded by obtaining new equipment.

Debottlenecking projects will involve better scheduling of process operations, improved allocation of existing equipment among the many products on the current product slate, and higher production rates for single products. These increased production rates would be brought about by making changes to process operating variables and operating times. Retrofitting projects will involve selective additions of equipment units. These problems will be quite complicated to solve for the general multipurpose plant. No current methods are available which can solve all aspects of either.

3.2 Designing a New Process

3.2.1 Typical Problem Statement

When a specialty chemical company develops a new product which it plans to add to its product slate, the new product will probably not be produced in a new grass roots batch plant for the reasons given above. Instead, the company will attempt to integrate the new process into an existing multipurpose plant. Making use of existing equipment will save money by reducing capital investment, so naturally existing units will be used exclusively if possible. A typical problem statement might be:

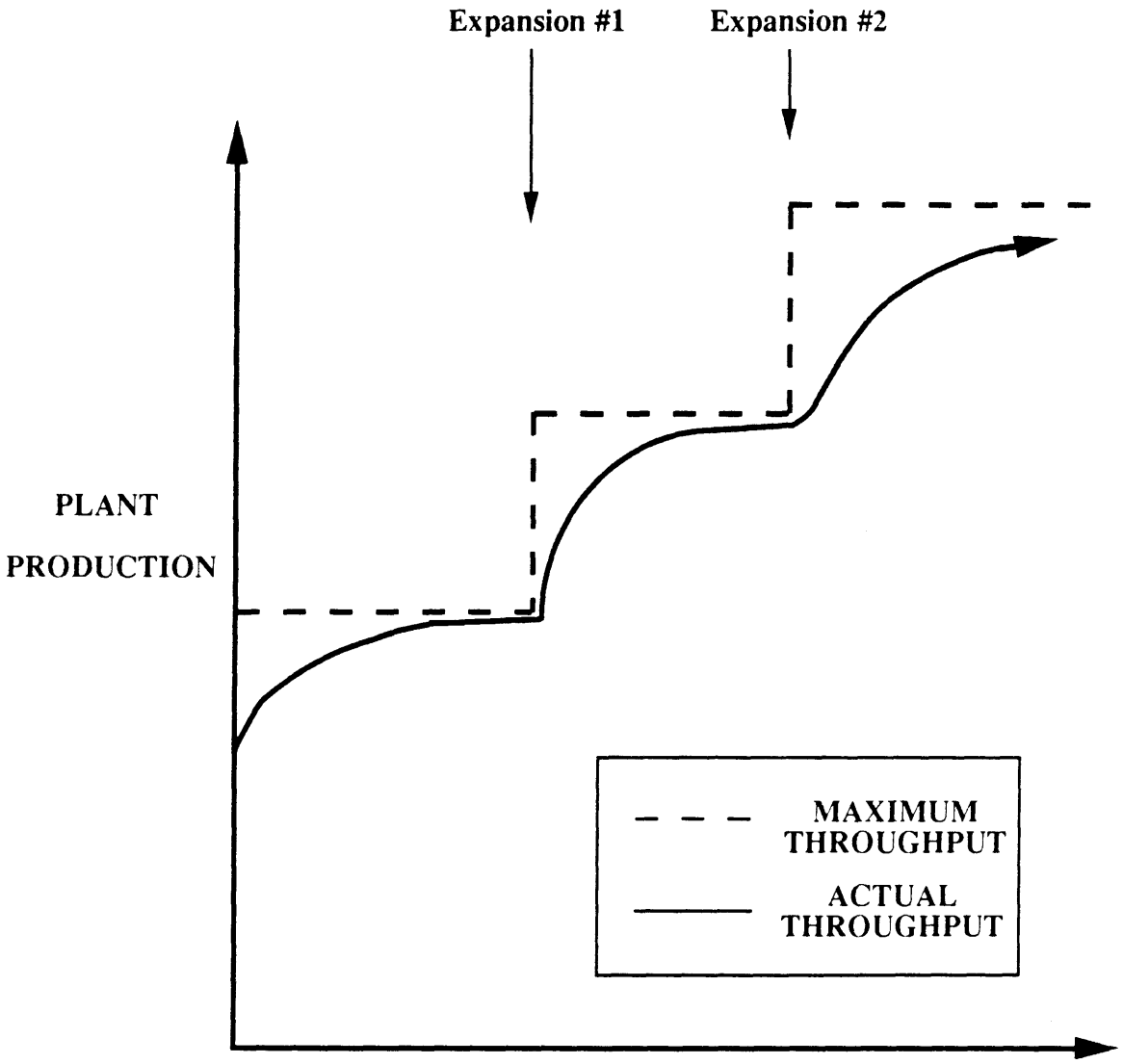


FIGURE 3-1. SCHEMATIC OF STAGED CAPACITY EXPANSION OF MULTIPURPOSE BATCH PLANTS

Management wants to make 10,000 kilograms of Product X for market evaluation. The new material has been successfully produced by chemists in the laboratory. How should the product be made in the plant? How much will it cost? And, how long will it take?

To carry out the design, decisions must be made in each of the five subtask design areas described in Section 2.2.1. In scaling the process from the lab bench or pilot plant to the production plant, these decisions must be made with some overall objective in mind in order to generate a "good" design. For the optimal sizing problem reported in the literature, this objective has generally been the minimization of capital cost. Since the new product will be produced in an existing batch plant, the minimization of capital cost does not apply as a primary design goal.

3.2.2 Choice of Objective Function

From an overall plant standpoint, the maximization of profit is the most likely overall goal. Profit from plant operations (Φ) could be given by an expression of the form:

$$\Phi = \sum_{p=1}^{P} (\phi_{Sp} - \phi_{Mp}) - \phi_L - \phi_O - \phi_E \qquad (3\text{-}1)$$

where ϕ_{Sp} is the total revenue from sales of product p, ϕ_{Mp} is the total direct cost of manufacturing product p (raw materials, energy, etc.), ϕ_L is the plant labor cost, ϕ_O is the total overhead cost, and ϕ_E is the total depreciation charge for equipment. Some terms are dependent on which products are produced and the amount of each, while some terms account for fixed costs for the plant as a whole. From an overall plant viewpoint, operating labor and process equipment units are resources which will be allocated to the production of a set of products at specified levels. These resources cost the same no matter how they are utilized.

The determination of the products to make and the amount of each to produce in the plant over some production horizon constitutes an upper level planning problem. Plant capacity constraints need to be incorporated into the problem. Also, limits on the minimum or maximum amount of production for each product (market share or demand) might exist. Depending on the plant utilization level, the objective of maximizing profit might manifest itself in different ways. In particular, the "optimal" process operating conditions could be affected.

If a batch facility producing multiple products is operating at or near full capacity, then the products that generate the highest margin per unit production time will be produced preferentially. Also, slight decreases in profit margins by producing individual products at higher average rates would be worthwhile if profit margin per unit time were increased. However, if the facility is under-utilized and all product demands are being satisfied, then maximizing profit corresponds to minimizing operating costs. In other words, each process should be produced as efficiently (highest margin) as possible.

The question then becomes how to convert this overall objective of maximizing profit into an appropriate objective for the design of a process to produce a single new product. Because the overall production planning problem is quite involved, a way to separate out just design of the new product is desired. However, some measure of the effects of the rest of the plant must be incorporated into the objective so that the single product is not optimized at the expense of the overall plant.

3.2.3 Allocating Fixed Costs

This problem partitioning goal is accomplished by allocating the fixed cost terms for the plant to the individual products. A number of possible allocation schemes are possible. The following method is used here. An implicit assumption in this method is that additional production would be scheduled on the plant if free time were available. Each fixed cost term is converted into an hourly charge rate. Depreciation and labor costs are divided by the total available production hours. Overhead charge rates are based on the total overhead costs divided by the total number of actual *product* production hours in some previous time period. Note that producing two products at the same time would count double. These hourly charge rates could then be thought of as rental charges which need to be paid in order to obtain production time.

From the plant standpoint, the fixed charges should be accounted for from the contributions of all the products. At the individual product level, the rental charge rates allow trade-offs between the efficiency and speed of processing to be quantified. For example, high yields may be possible for product A at the expense of a long reaction time. A shorter reaction time (lower yield) might increase raw materials cost slightly for product A, but free up enough reactor time that another profit producing product could be run on the reactor. Thus, the rental charge could be thought of as the opportunity cost of allocating an equipment unit to one product as opposed to any other.

The rental charges might be determined in some fashion other than that described above. For example, the plant floor might be thought of as a market where products compete for scarce resources (equipment units). The rental charges would represent the prices of the scarce goods. Equipment units in high demand would command high prices. This pricing system would encourage products to use lower rent units if possible. The setting of the relative rental prices might represent a way to allocate the equipment units to the best set of products for the plant.

It is important to remember that the use of rental charges is simply a way to break the overall plant optimization problem down into more manageable parts. In addition, if the assumption that the plant is nearly fully utilized is not correct, then the individual products should be produced as efficiently as possible. There is no point in producing material faster than necessary if idle time will remain for the plant after all production has been completed.

By allocating the fixed charges among the products through the use of rental charges, the design objective for a new product is then also the maximization of profit. If the amount of product to be produced is specified, then maximizing profit is equivalent to minimizing total operating costs, where the allocated fixed costs appear as rental or equipment usage charges.

3.2.4 Design Assumptions

A process is assumed to consist of a set of tasks which are to be carried out in a specified order. These tasks are classified by type (e.g., reaction, distillation, etc.) and have been assigned to processing stages of the same type. The problem of assigning recipe tasks to process stages (the task merging/splitting problem) is assumed to have been carried out already.

A subset of the equipment in the plant is available for use in making the new product. This "inventory" consists of various types of units, which can be freed up by the plant scheduler for use with this new product. However, not all of the units in the inventory need be used in producing the new product. Unused units would remain available to other products in the plant. The method by which units are placed in the equipment inventory is outside the scope of this work. A one-to-one matching of task types to equipment types is assumed to exist. For each unit type, there are a number of

discrete sizes available with one or more units at each size. Also, each unit has an equipment usage charge (rental charge) associated with it.

3.3 Formulation of Design Problem

3.3.1 Differences from Previous Work

The design problem we are formulating differs from most of the previous formulations presented in the literature in three major respects:

(1) The processing time in each piece of equipment is not necessarily fixed. Instead, constraints on process performance are specified. The processing times and operating conditions in the equipment units can be varied to optimize the overall cost of the process while satisfying these performance requirements.

(2) The process takes place in an existing plant rather than in a new grass roots plant. Equipment units are selected from a discrete set of available equipment items rather than purchased from a continuous range of sizes.

(3) A more appropriate objective function than the minimization of initial capital investment is the minimization of the total manufacturing cost, including charges for the use of equipment, raw materials, energy, and labor.

These differences complicate the design problem, and make solution considerably more difficult.

Note also that when a product is to be manufactured using existing equipment in a batch plant, there is not a clear distinction between the problems of process design and process operation. The "design" of a batch process typically consists of deciding which equipment should be used and then how the equipment should be operated to make the product. In addition, when a new product is produced in an existing plant, the problem is by definition a "retrofit" design problem.

3.3.2 Given Information

A number of simplifying assumptions are made in terms of the process operating mode in order to focus on performance issues. The problem formulation for the optimal

design of multiproduct batch plants considered in this thesis starts with the following information:

(1) A slate of P products is to be produced in a series of single-product campaigns. The quantities of each product to be produced over a global time horizon H are specified.

(2) An inventory of J types of equipment units is available. For each unit type, there are K_j units available. These units have known sizes.

(3) The general process flowsheet for all the products is assumed to be given. For each product, all of the required processing tasks have been identified and assigned to an appropriate unit type. Task merging and splitting is assumed to have been finished. Thus, the process synthesis and task to stage assignment problems have been completed. The process is assumed to use periodic, overlapping operation.

(4) An inventory of intermediate storage tanks is available. The number and sizes of the tanks are specified.

(5) There is a rental cost per unit time charged for the use of each equipment item and each storage tank. The equipment rental charges could be based on depreciation to recover the initial capital investment or on the internal demand for the units. In either case, the rental rates are assumed to be given for this analysis.

(6) Mathematical models are available to calculate the performance of each processing unit as a function of operation time, inputs from previous stages, and specified operating variables, such as reaction temperature and distillation reflux ratio. The operating variables may also be functions of time. In these cases, the operating profile can be approximated as a set of piecewise constant sections.

(7) The policy for providing intermediate storage between successive stages in the process is specified as no intermediate storage or unlimited intermediate storage (with equal average rates of all subtrains).

The design problem consists of selecting appropriately sized process and intermediate storage units from the available equipment inventory, choosing the operating times and the profiles of the operating variables for all the units for all the products, and distributing the available horizon time H among the P products. The processes must produce on-specification products, and the campaigns should be completed within the given time horizon. In addition, these decision variables should be determined in order to minimize the total cost of manufacturing all the products. The manufacturing cost can include raw materials, utility consumption, equipment rental, operating labor, and cleaning or set-up charges.

This formulation could be modified to handle the purchase of new units or the design of a new grass-roots plant. If equipment is purchased, we follow the lead of Sparrow et al. (1975) and assume that it is only available in standard sizes from vendor's stock. Some mechanism would need to be incorporated to distinguish between alternatives that do not require new units and those that do. The rental rate for newly purchased equipment might be set as some form of adjusted charge for depreciation. The adjustment might take the form of a multiplicative penalty factor to account for the increased risk associated with additional investment.

3.3.3 MINLP Formulation

The general optimization problem for the design of a process for producing a new product using existing equipment can be formulated as a mathematical programming problem. The objective function, ϕ, is the minimization of total manufacturing costs:

$$\phi = \sum_{p=1}^{P} \left(F_p\left(\underline{x}_p, \underline{t}_p, \underline{u}_p, \underline{B}_p\right) A_p \; + \; E(\underline{y})\, \tau_p\right) \qquad (3\text{-}2)$$

The objective function consists of two types of terms. The first term is the direct manufacturing cost for producing product p, which depends on the efficiency of production and the amount produced. This term would include the raw materials, waste treatment, and utilities costs. F_p is some general nonlinear function of the operating times, $\underline{t}_p$, process operating variables, $\underline{x}_p$, dependent variables, $\underline{u}_p$, and stage batch sizes, $\underline{B}_p$, which gives the cost per unit amount of product p produced. The production requirement for product p is given as A_p. The second term represents the allocated fixed costs for overhead, equipment

depreciation, and labor. This term is dependent on the rental cost function, E, the choices of units from the equipment inventory, $\underline{y}$, and the campaign time for product p, τ_p.

The solution of the optimal design problem requires a model of the process that can calculate the objective function as a function of the decision variables. A sequential simulation approach is taken here to model the process. Key input variables (i.e., operation times ($\underline{t}_p$) and operating variables ($\underline{x}_p$)) are specified for each unit. Dependent variables ($\underline{u}_p$), such as mole fractions, temperature, or other performance measures, are supplied with the appropriate initial conditions and calculated as a function of the decision variables. The calculations flow from the first unit through each succeeding unit. Since the unit models are typically time dependent, the simulation will require the integration of ordinary differential equations and the solution of systems of nonlinear algebraic equations in the general case. Material recycle is assumed to occur through external storage units in which the properties of the recycled material are known.

The process model for product p can be represented by a set of algebraic and differential equations relating the dependent variables to the decision variables.

$$\underline{u}_p = \underline{f}_p\,(\underline{x}_p,\underline{t}_p,\underline{u}_p,\underline{B}_p) \qquad \text{for } p=1,P \qquad (3\text{-}3)$$

$$\frac{d\underline{u}_p}{dt} = \underline{g}_p\,(\underline{x}_p,\underline{t}_p,\underline{u}_p,\underline{B}_p) \qquad \text{for } p=1,P \qquad (3\text{-}4)$$

After simulating the process for each product, size factors (S_{ijp}) are calculated for each stage i of type j for each product p for the given values of decision variables.

$$S_{ijp} = h_{ijp}\,(\underline{x}_p,\underline{t}_p,\underline{u}_p,\underline{B}_p) \qquad \text{for all } i,j;\ p=1,P \qquad (3\text{-}5)$$

The process model and equations for calculating the size factors must be provided by the user since they are problem specific.

There are a large number of additional constraints that must be satisfied. These include constraints on the logical variables which are used to specify the unit to stage assignments and relationships required to determine the average production rate. Each of these sets of constraints is described below.

The process is assumed to consist of J types of stages, with I_j stages of type j. In addition, the equipment inventory is assumed to contain K_j units of type j. A binary

variable, y_{ijk}, is used to indicate the assignment of unit k of type j to stage i of type j.

$$y_{ijk} \in \{0,1\} \qquad \text{for all i,j,k} \qquad (3\text{-}6)$$

The binary integer variables introduce combinatorial aspects to the optimization problem. The total number of integer variables is equal to:

$$N_{bvar} = \sum_{j=1}^{J} I_j K_j \qquad (3\text{-}7)$$

The number of parallel units, m_{ij}, present at stage i of type j is given by:

$$m_{ij} = \sum_{k=1}^{K_j} y_{ijk} \qquad \text{for all i,j} \qquad (3\text{-}8)$$

Since a unit k can only be used in at most one location at a time, the following constraints are imposed on the binary variables:

$$\sum_{i=1}^{I_j} y_{ijk} \leq 1 \qquad \text{for all j,k} \qquad (3\text{-}9)$$

Also, each processing stage i must be assigned at least one unit.

$$\sum_{k=1}^{K_j} y_{ijk} \geq 1 \qquad \text{for all i,j} \qquad (3\text{-}10)$$

Despite these feasibility constraints on the binary variables, the total number of possible structures grows explosively as the number of processing stages and the number of available units increase.

The set of constraints for determining the average production rate depends on the storage policy chosen for the plant. The following constraint set is formulated for the case when storage is included between all stages in the process. The required storage volumes are estimated by Equation 2-12 given in Section 2.1.4. The cycle time (excluding idle time) for stage ij for product p with unit k assigned, T_{ijkp}, is given by:

$$T_{ijkp} = t_{ijp} + \alpha_{ijp} + \beta_{ijp} B_{ijkp} \qquad \text{for all i,j,k; p=1,P} \qquad (3\text{-}11)$$

where t_{ijp} is the processing time for stage ij for product p, α_{ijp} is a fixed transfer or set-up time, and the β_{ijp} term incorporates size-dependent change-over times. The batch size of stage ij for product p with unit k assigned is denoted by B_{ijkp}. The batch size constraint takes the following form:

$$y_{ijk} V_{jk} \geq S_{ijp} B_{ijkp} \qquad \text{for all i,j,k; p=1,P} \qquad (3\text{-}12)$$

where V_{jk} is the volume of unit k of type j. If a unit is not utilized, y_{ijk} takes a value of zero, and Equation 3-12 forces the value of B_{ijkp} to zero as well.

Once the cycle time and batch sizes have been determined by the constraints above, the average production rate for product p for unit k at stage ij, r_{ijkp}, can be calculated:

$$r_{ijkp} = \frac{B_{ijkp}}{T_{ijkp}} \qquad \text{for all i,j,k; p=1,P} \qquad (3\text{-}13)$$

The average production rate for stage ij, R_{ijp}, is simply the sum of the rates of the units assigned to that stage:

$$R_{ijp} = \sum_{k=1}^{K_j} r_{ijkp} \qquad \text{for all i,j; p=1,P} \qquad (3\text{-}14)$$

The limiting average rate for each product p, R_{Lp}, is given by:

$$R_{L_p} = \min_{\forall\, i,j} \{R_{ijp}\} \qquad \text{for p=1,P} \qquad (3\text{-}15)$$

The campaign time for product p, ignoring end effects, is then:

$$\tau_p = \frac{A_p}{R_{L_p}} \qquad \text{for p=1,P} \qquad (3\text{-}16)$$

The overall horizon constraint for multiproduct operation is:

$$\sum_{p=1}^{P} \tau_p \leq H \qquad (3\text{-}17)$$

This constraint ignores product change-over times, which should be negligible since long campaigns are being produced.

Finally, there may be simple bound constraints on the variables in the MINLP.

$$\underline{x}_p^L \le \underline{x}_p \le \underline{x}_p^H \qquad \text{for } p=1,P \qquad (3\text{-}18)$$

$$t_{ijkp}^L \le t_{ijkp} \le t_{ijkp}^H \qquad \text{for all } i,j,k;\ p=1,P \qquad (3\text{-}19)$$

$$\underline{u}_p^L \le \underline{u}_p \le \underline{u}_p^H \qquad \text{for } p=1,P \qquad (3\text{-}20)$$

$$B_{ijkp}^L \le B_{ijkp} \le B_{ijkp}^H \qquad \text{for all } i,j,k;\ p=1,P \qquad (3\text{-}21)$$

The entire MINLP for designing a multiproduct process with intermediate storage between all stages consists of minimizing the objective function given in Equation 3-2 subject to the constraints given by Equations 2-12, 3-3 to 3-6, and 3-9 to 3-21. The y_{ijk} are the discrete variables, and $\underline{x}_p$, t_{ijkp}, $\underline{u}_p$, B_{ijkp}, T_{ijkp}, r_{ijkp}, S_{ijp}, R_{ijp}, R^L_p, and τ_p are the continuous variables.

This mathematical programming problem is quite difficult to solve for a number of reasons. First, the discrete variables cause the problem to be combinatoric. The discrete variables also occur in nonlinear constraints. This condition makes it more difficult to apply MINLP solution methods. Second, the formulation involves a large number of optimization variables. Also, the continuous relaxation of the problem is very likely non-convex in most cases. Thus, a unique minimizer is not guaranteed, and the nonlinear problem could be difficult to converge. Finally, the process models for the calculation of the size factors typically involve the numerical integration of differential equations, which can be quite computationally demanding.

3.4 Decomposition Strategy

A decomposition strategy is proposed for attacking the overall design optimization problem. The complete set of decision variables is partitioned into two subsets. This partitioning generates two subproblems, which can be studied both individually and together. The first subset of decision variables consists of the continuous variables for the processing times and operating variables. The discrete variables that determine the process structure make up the second set.

The main motivation for pursuing a decomposition approach is that by breaking the overall problem up into smaller parts, it should be easier to get an understanding of the fundamental nature of the problem. By studying the smaller, more manageable subproblems first, it will be easier to identify the more complex interactions found in the general problem. In addition, this problem presents a very natural partitioning of the optimization variables. The method of approach followed in this thesis is to start with simple problems, gain experience and understanding, and gradually move on to solve more complex problems.

By fixing the process structure (discrete variables), the combinatorial aspects of the problem are removed and the "performance subproblem" is generated. This subproblem consists of finding the best process operating conditions and times for a process with fixed equipment. The problem of determining optimal process performance for multistage batch processes with fixed equipment has not been extensively studied in the batch process design literature. This continuous optimization problem is studied in Chapter 4.

By fixing the process operating variables and times (continuous decision variables), the "structure subproblem" is generated. Almost all of the batch process design literature has dealt with problems with given performance. This version of the structure subproblem differs in the form of the objective function and the fact that existing units are being used. The structure subproblem is examined in Chapter 5.

The two subproblems are studied individually in order to get a better understanding of the fundamental aspects of these particular subproblems. The complete design problem, when these two subproblems must interact in an overall solution strategy, is discussed in Chapter 6. The entire optimization problem consists of solving a nested set of subproblems. The appropriate nesting of the two subsets of the decision variables as well as the key issues in developing an overall solution approach are considered.

Chapter 4

PERFORMANCE SUBPROBLEM

4.1 Introduction

The performance subproblem deals with only task (b) of the design tasks described in Section 2.2.1. Decisions are assumed to have been made already for the other four tasks, and thus the overall operating mode, the task to stage assignments, the number of parallel units at each stage, the intermediate storage locations, and the unit sizes have been fixed. The only trade-offs left to be investigated concern the choices of process operating variables and operating times.

In many ways the performance subproblem is an "optimal operations" problem, mainly because the process operating behavior is the focus of the optimization and because the equipment sizes are fixed. Solution of the performance subproblem might be necessary during the design of a new process or the debottlenecking of an existing one. In the former case, process information from the lab or pilot plant would be used to scale up the process. The latter case involves the optimization of an existing process in order to improve it in some way. The goal of this performance optimization might be an increase in average production rate, a reduction in operating costs, or an improvement in product quality. In both cases, constraints imposed by fixed equipment affect the choices of operating variables and times. Also, the current overall plant environment can affect the optimization through the choice of objective function and the values of the equipment rental charges.

This chapter covers the performance subproblem. First, the generic types of trade-offs that must be considered to optimize process performance are identified and described. Next, the overall mathematical problem formulation from Chapter 3 is simplified to handle this subproblem. A solution approach is then described. Simple example problems are solved to demonstrate the various trade-offs and show the application of the solution approach. An alternative approximate approach that reduces the computational effort required when solving multiproduct performance subproblems is also

described. Finally, some important implementation issues for the application of these solution approaches to larger problems are discussed.

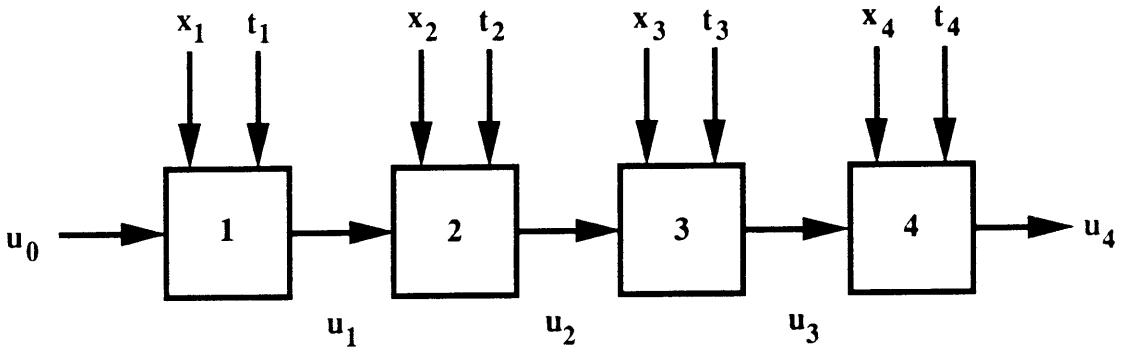
4.2 Performance Trade-offs

A number of generic performance trade-offs exist for the design of a batch plant. In general, however, these trade-offs essentially always involve to some extent the question of processing efficiency versus processing speed (average production rate). If an infinite supply of process units were available at no charge, then the best design would be the one that produced the product most efficiently, that is, with least operating cost. Any constraint on minimum production rate could be satisfied by adding more equipment units in parallel. Since the additional equipment is free, there would be no increase in cost per amount of product.

In reality, equipment units are generally not available in infinite numbers and are certainly not free. The total production requirement and the time horizon for the product determine a minimum average production rate which must be satisfied by the process. If the most efficient design satisfies this minimum production rate and the plant has excess capacity, the most efficient design would still be the best. However, if the equipment units are heavily utilized, it may be worthwhile to design a process that has a higher average production rate. This higher production rate would also probably require less efficient processing and thus have higher processing costs (excluding equipment usage charges). However, the shorter campaign time requirement might be better for the plant as a whole because more products could then be made on the equipment units.

In order to incorporate the overall plant production environment into the optimal design of a single new product, rental or equipment usage charges were introduced in Section 3.2.3. These rental charges are an attempt to quantify the value of the equipment so that this fundamental trade-off of processing efficiency versus processing speed can be incorporated into the optimization problem. Since the equipment units are fixed for the performance subproblem, the total cost associated with the equipment will increase linearly with the overall campaign time. The campaign time is dependent on the choices made for the processing times and the process operating times. These decisions also affect the efficiency of processing and hence the rest of the operating costs.

Consider the multi-stage batch process shown in Figure 4-1. Between each pair of stages, a vector of performance attributes ($\underline{u}_i$) is shown. These performance attributes are



u_i Vector of Performance Attributes for Stage i

x_i Vector of Process Operating Variables for Stage i

t_i Vector of Processing Times for Tasks at Stage i

FIGURE 4-1. BLOCK DIAGRAM OF FOUR STAGE BATCH PROCESS

dependent on the initial conditions ($\underline{u}_0$), the operating times at each stage ($\underline{t}_i$), and the process operating variables ($\underline{x}_i$). There are two general types of performance trade-offs which will be discussed in this section. The first type considers the case when the interstage performance attributes ($\underline{u}_i$) are given specified values. In the second case, these values are allowed to vary.

The first type of trade-off, processing intensity versus unit cycle time, occurs within a single unit. Process operating variables ($\underline{x}_i$) and times ($\underline{t}_i$) are varied such that the specified endpoints ($\underline{u}_i$) are obtained. Increasing the intensity of processing (e.g., higher temperature, higher catalyst loading, more vigorous mixing, etc.) reduces the time required to reach the required performance specification. The higher cost associated with more intense processing would be weighed against reduced production time requirements.

Even though the interstage performance attributes are fixed, each stage cannot be considered independently. The location of the stage with the bottleneck average production rate determines whether proposed processing modifications are worthwhile. If a stage is not the bottleneck, there is no benefit to increasing the processing intensity (and the processing costs) because reducing the cycle time of a non-bottleneck stage has no impact on the overall average rate. Thus, the strategy for non-bottleneck stages would be to operate as efficiently as possibly while not becoming the bottleneck. The goal for the bottleneck stage would be to increase the average rate as long as the stage remains the bottleneck and the increased operating costs are offset by reduced equipment costs.

For example, consider a two stage process with a batch reactor followed by a batch distillation column. If the reactor is the bottleneck stage, reducing the time required to reach a specified conversion would speed up the entire process. Running the reactor at a higher temperature might be one way to achieve this goal. Reducing the reactor cycle time would reduce the campaign time, and the total cost for labor and equipment would decrease. These savings would have to be weighed against increased energy costs.

The second type of performance trade-off occurs between or among units. In this case, the interstage performance attributes ($\underline{u}_i$) are not specified. By allowing these values to float, the "performance load" can be distributed between the units to take advantage of cost factors, such as rental rates, utility costs, etc., and the relative time requirements of the stages. Although some performance specifications (e.g., final conditions) may be fixed, the remaining ones would be determined by setting the operating variables ($\underline{x}_i$) and times ($\underline{t}_i$) in order to obtain a global optimum.

Performance load trade-offs involve the extent to which the processing tasks are carried out in each stage. By making an operating change at one stage, some adjustment may need to be made downstream in the process in order to satisfy final performance specifications. By adjusting the load at each stage, the speed and efficiency of processing can be balanced throughout the process.

Consider again the two stage process with batch reaction followed by batch distillation. Suppose that the reaction conversion was initially specified at 95 percent, and the time requirements of the two stages were such that the reactor was the bottleneck. If the separation in the column was fairly easy, it might be worthwhile to reduce the conversion in the reactor. This change would decrease the reaction time and shift more of the processing load to the column. However, an easy separation might mean that the distillation time would not increase much. The increased overall production rate and resulting lower equipment costs might offset the increased energy costs and lower yield on raw materials.

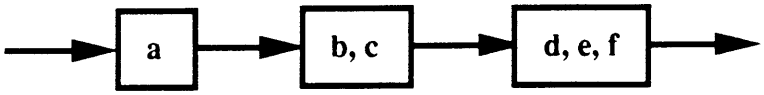
One might consider the task to stage assignment problem (Yeh and Reklaitis, 1987) to be one where performance load trade-offs are involved. In this problem, tasks with short time requirements can be merged together. This merging reduces the number of stages (units). Although the performance is fixed for each task (given task size factors), changing the number of tasks assigned to a stage is like changing the extent of processing. For a specified number of stages, the stages are merged to balance the stage time requirements, or in other words, the performance load. Figure 4-2 illustrates two possible task to stage assignments for a case with six tasks and three stages. Alternative B is better balanced because task (d) has been moved to stage 2. This change reduces the limiting cycle time and thus increases the average production rate.

These two types of general performance trade-offs need to be considered in order to design an optimal batch process. Because the overall plant environment affects the optimal values for the operating variables and processing times, the processing conditions used by the chemist at the laboratory scale may not necessarily translate to the most economically desirable conditions at full production scale. Performance trade-offs must be considered on an overall basis in order to obtain a global optimum.

PROCESS DATA

Tasks	a	b	c	d	e	f
Time Requirements (hr)	4	2	2	1	3	2

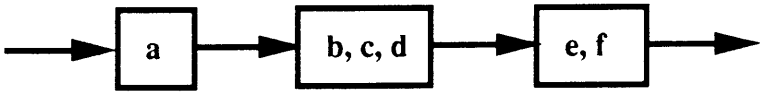
ALTERNATIVE A



Stage Cycle Time 4 hrs 4 hrs 6 hrs

Limiting Cycle Time = 6 hrs

ALTERNATIVE B



Stage Cycle Time 4 hrs 5 hrs 5 hrs

Limiting Cycle Time = 5 hrs

FIGURE 4-2. TASK MERGING AND SPLITTING AS A PERFORMANCE LOAD TRADE-OFF: COMPARISON OF TWO POSSIBLE TASK TO STAGE ASSIGNMENTS

4.3 NLP Formulation

It is desirable to explore these performance trade-offs in a systematic fashion. By fixing the units in the process (y_{ijk} values), the MINLP posed in Section 3.3.3 can be simplified to a nonlinear programming problem (NLP). This NLP formulation for the performance subproblem has the operation times ($\underline{t}$) and the operating variables ($\underline{x}$) as the decision variables. For a fixed production requirement, the objective function (given by Equation 3-2) consists of minimizing the operating costs, including the raw materials, waste treatment, utilities, and equipment usage costs.

The initial conditions and decision variables serve as inputs to the process model which generates outputs by solving the appropriate differential and algebraic model equations. The process model equations take the form of constraints given by Equations 3-3 to 3-5. Additional constraints for determining cycle times, batch sizes, and average production rates are given by Equations 3-11 through 3-17. The actual number of constraints is greatly reduced because the y_{ijk} values are known once the structure is fixed. Also, the number of variables is reduced because batch sizes, cycle times, and average rates are needed only for existing equipment to stage assignments rather than for all possible assignments. Variable bounds are given by Equations 3-18 to 3-21.

For any design problem there are two main goals. First, find a feasible solution if one exists. Second, find an optimal solution based on the criteria established by the objective function. The classic batch design problem addressed in the literature consists of minimizing the initial capital investment for a batch plant for the case when the process recipe was fixed. Thus, performance constraints were assumed to be satisfied implicitly and did not present any obstacle to finding a feasible solution. Also, continuous size ranges for equipment units were generally considered. Units could always be made bigger or added in parallel to insure that the average processing rate constraint was satisfied. Thus, for this class of problems, investigators did not have to be concerned with feasibility but only with optimality.

The batch design problem described here, which includes the effects of performance, has a greater problem with the feasibility issue. Performance specifications (such as purity requirements) may not be attainable for a proposed processing system. Also, since unit sizes are given, it may be impossible to satisfy the campaign horizon constraint. Feasibility is a primary concern for engineers scaling up or designing a new

process. The NLP described above could be modified to get at least one feasible solution by replacing the minimum cost objective function with the sum of the squares of the constraint violations. The new NLP would be solved to get an initial feasible solution. This feasible solution could then be used as the initial guess for solving the minimum cost NLP optimization problem.

4.4 Solution Approach

The NLP problem formulated for the performance subproblem (Equations 3-2 to 3-5 and 3-11 to 3-21) can be solved by using a successive quadratic programming (SQP) algorithm. Pike (1986) provides a general description of SQP algorithms as well a number of good references. Fletcher (1981a,b) is another good source of general information on optimization methods for NLP's. A brief outline of how the SQP algorithm works is given here.

First, an approximate quadratic programming problem (QP) is constructed about an initial trial point. This QP is solved to obtain a search direction. The quadratic objective function for the QP is obtained by making a second order approximation of the nonlinear objective function about the trial point. Linear equality constraints for the QP are obtained by linearizing the nonlinear constraints and adding slack variables to all inequality constraints. The optimal point obtained by solving the approximated QP and the initial trial point are used to calculate the search direction. A line search is then performed along this search direction to reduce the objective function. The new point obtained from the line search is used as the starting point for the next QP. These steps (QP problem, line search) are repeated until an optimal solution is found.

The NLP problem requires that a process model be available to describe the performance of the process in response to changes in the values of the operating times and process operating variables. A black box approach is used in this thesis to handle the process model and any general equality constraints. Key variables are identified such that the process behavior can be calculated in a straight-through fashion once these key inputs are given values. Essentially, values for the key inputs are provided to the subroutine for the process model on each iteration, and the model returns outputs. These outputs include the objective function value and the value of all nonlinear inequality constraints.

This black box approach reduces the number of variables that the optimization routine has to handle because the process model (and all equality constraints) are

automatically satisfied on every iteration. Also, the integration of any differential equations is carried out inside the black box for the process model. Derivatives for the objective function and nonlinear inequality constraints are calculated by finite difference approximation, which requires one re-simulation for each decision variable perturbed.

There are some potential numerical problems that may be encountered when solving the NLP. First, proper scaling of the optimization variables, the objective function, and all constraints is known to have an important effect on the performance of the SQP optimization algorithm. Because detailed models are being used, variable scaling may not be a trivial exercise. By using the black box method described above, only the key decision variables, the objective function, and the inequality constraints need to be scaled. The main objective of scaling is to generate scaled variables and constraints which are all of order one.

Second, the feasible region generated by the constraints in the problem may be non-convex. In fact, with the complexity of the models to describe process performance, it would be surprising if non-convexities were not encountered. The presence of non-convexities means that there is no guarantee that an optimal solution will be obtained by the SQP algorithm. The SQP method may converge to a local (but non-global) optimum.

There are a few ways to try to deal with non-convexities in the feasible region. One way is to try to convexify the equations on a case by case basis and solve a relaxed version of the actual problem. Bounding methods might be used to determine the global optimum. A second approach is to solve the non-convex NLP for a number of different starting points and take the best solution. The hope is that a near-optimal solution can be found without excessive computational effort.

4.5 Example Problems

Three example problems are solved to demonstrate the use of the SQP algorithm for solving the performance subproblem. Rather simple example problems have been devised in order to show the important aspects of the performance subproblem without getting lost in a mass of details. Our objective is the development of an understanding of the fundamental nature of the problem class and the important trade-offs.

All three problems involve two stage processes. The first problem is the simplest of the three and demonstrates a performance load trade-off. The second problem demonstrates a processing intensity trade-off as well as the interaction between the two

trade-off types. More complicated process models are used in the third problem in order to show their effects on the optimization method in terms of solution accuracy and computation time.

4.5.1 Example Problem #1

The first example problem demonstrates a performance load trade-off. The design of a process for the production of a single product is considered. The process consists of two stages with unlimited intermediate storage between them as shown in Figure 4-3. The reaction time is essentially the only independent decision variable in the problem. The reaction conversion is the interstage performance measure of interest and is a function of only the reaction time. A brief overview of the process is given below. The model equations are provided in Appendix A.1 for the interested reader.

A batch reactor of volume V_r converts reactant A to product B. Product B also degrades to form byproduct C. Both reactions are first order with given rate constants k_1 and k_2. The mole fractions of A, B, and C are determined as a function of reaction time for a reactor initially charged with pure A. Typical reactor concentration profiles are shown in Figure 4-4. At the end of the reaction cycle, the reactor contents are transferred to the intermediate storage. The reactor requires t_{cr} hours for draining, cleaning out, setting up for the next batch, and filling.

The storage tank feeds a batch distillation unit with a still capacity V_c. The column is assumed to operate such that it produces perfect splits between pairs of components. Because reactant A is the most volatile component and byproduct C is the least volatile, the assumption of perfect splits means that pure reactant A is removed first by the column. Then pure product B would be collected next. Typical still and distillate profiles are shown in Figure 4-5. The perfect splits assumed for this example correspond to a limiting case of infinite stages and total reflux.

The actual profiles would depend on the relative volatilities (or, more generally, on the phase equilibrium relationships) of the components involved, the number of stages in the column, the reflux ratio as a function of time, and the still pot composition. The column performance has been effectively decoupled from the operating variables that determine actual column performance for this example. This decoupling implicitly assumes that there are sufficient stages and that the reflux ratio chosen is high enough for the given conditions to approximate perfect splits. The limiting case represents a suitable starting

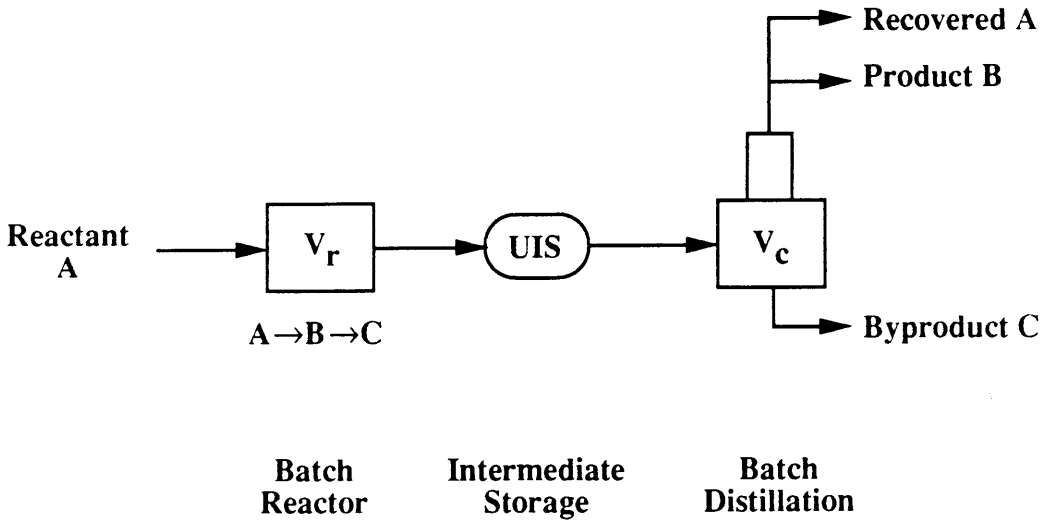


FIGURE 4-3. PROCESS FLOWSHEET FOR EXAMPLE PROBLEM #1

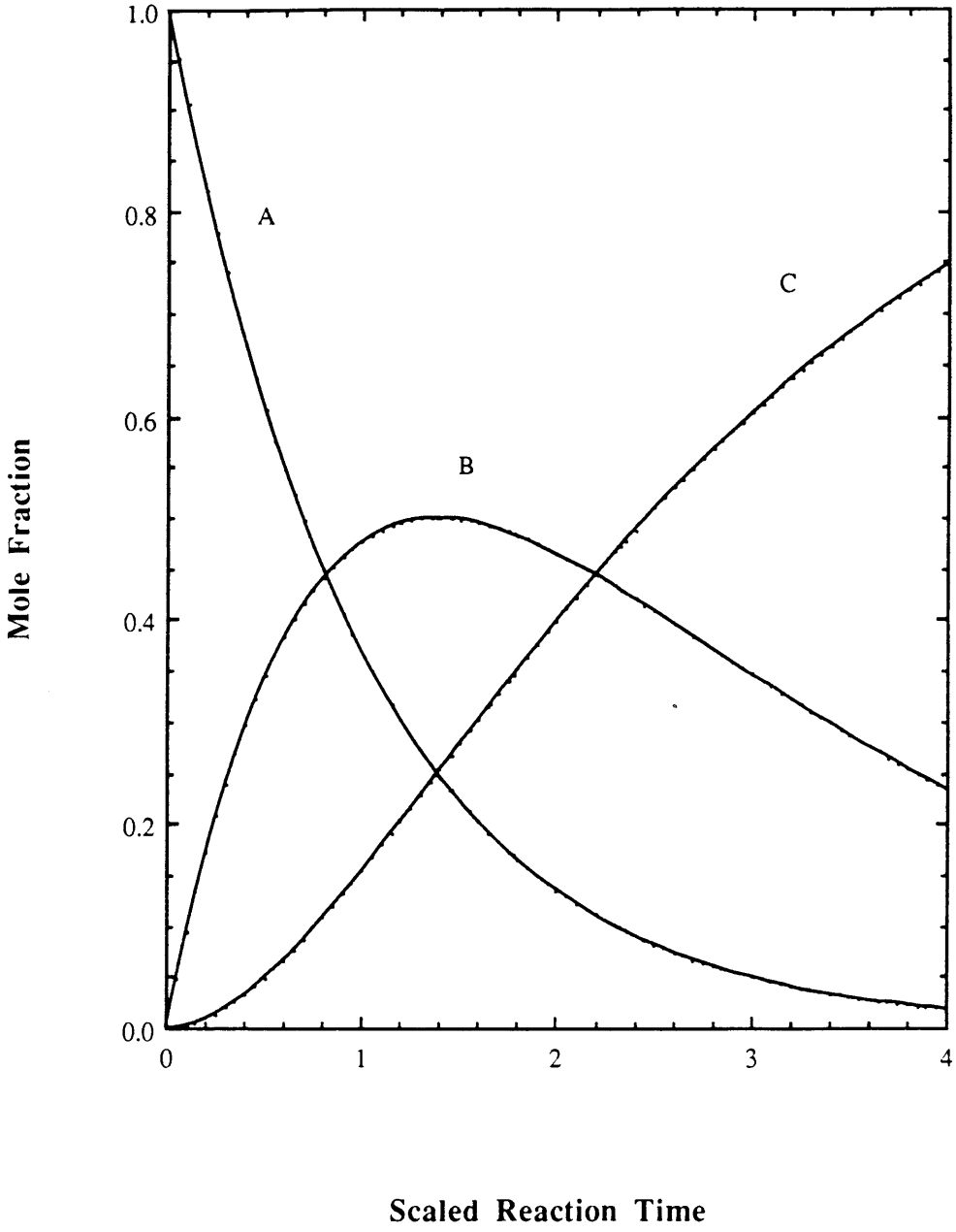


FIGURE 4-4. TYPICAL COMPOSITION PROFILES FOR BATCH REACTOR

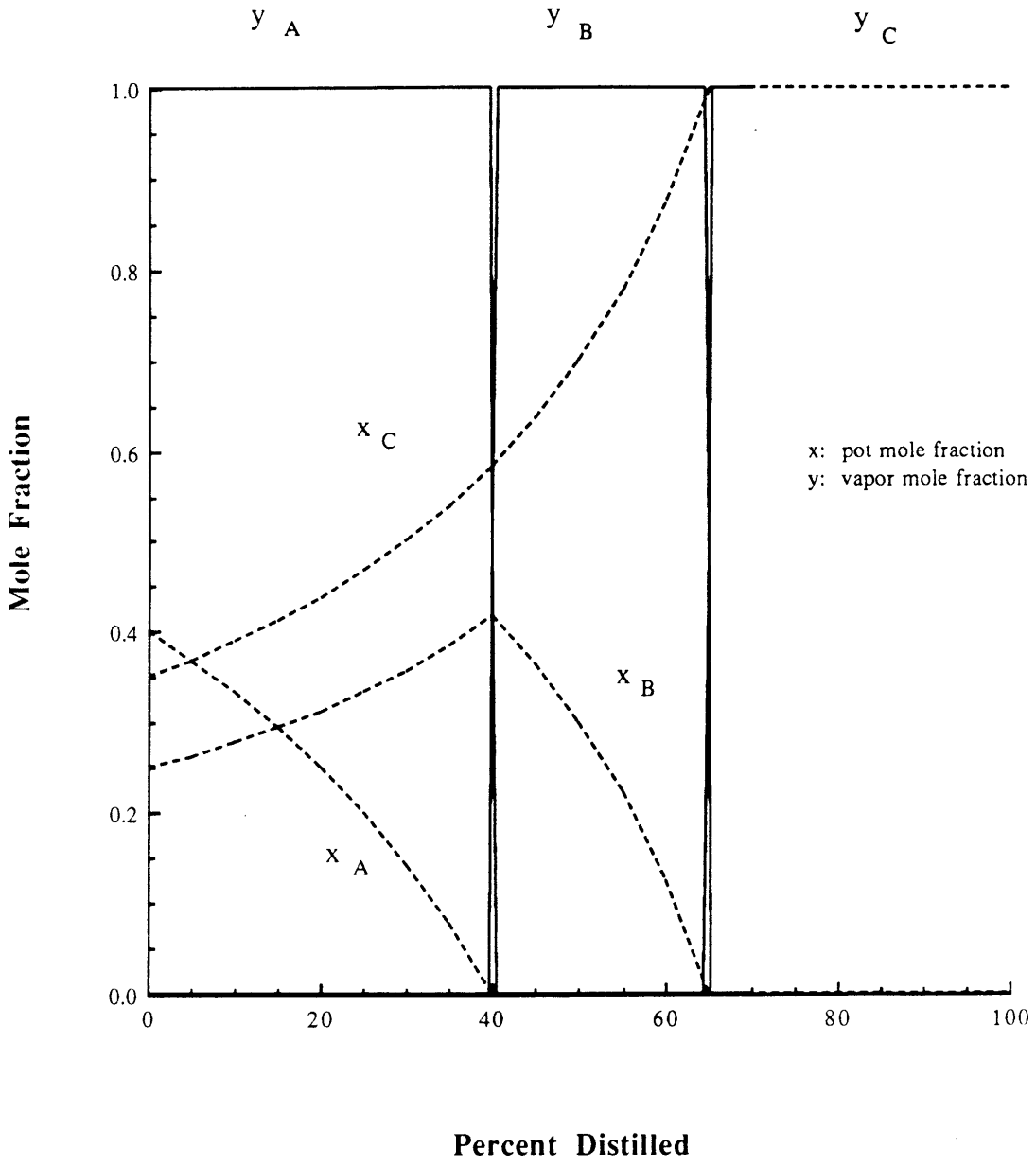


FIGURE 4-5. TYPICAL COMPOSITION PROFILES FOR BATCH DISTILLATION WITH PERFECT SPLITS

point for this example problem; it can be modified to describe a real column with finite stages with the reflux ratio as a decision variable. In addition, the computational requirement for modeling the column is greatly reduced with the perfect splits assumption.

The optimization objective function is the minimization of the total cost of a campaign to produce a required amount of product, B_{tot}. The total manufacturing cost consists of raw material, equipment rental, cleaning, and utility components. The entire NLP consists of a nonlinear objective function, six equality constraints (three linear, three nonlinear), three linear inequality constraints, and bound constraints for the eight optimization variables. The model requires nineteen user supplied parameters.

The optimization problem described above can be reduced in size by using the equality constraints to eliminate optimization variables. The problem can also be recast into a dimensionless form. The objective function is scaled in terms of a characteristic cost while the two remaining decision variables (reaction time t_r and the column cycle time T_c) are scaled by characteristic times for those units. Scaling the problem reduces the number of input parameters by forming dimensionless groups and makes the magnitudes of the decision variables order one. This scaling also improves the efficiency of the optimization algorithm solving the NLP.

The dimensionless formulation for the optimization problem is given below in general form:

$$\min \phi = \phi\,(t_r^*,\ T_c^*,\ N_{rxn},\ N_A,\ N_C,\ N_U,\ N_{clean},\ N_{cap}) \tag{4-1}$$

$$g_1(t_r^*,\ T_c^*,\ N_{rxn},\ N_{rate},\ N_{chr}) \geq 0 \tag{4-2}$$

$$g_2(t_r^*,\ T_c^*,\ N_{rxn},\ N_{chc}) \geq 0 \tag{4-3}$$

$$g_3(t_r^*,\ T_c^*,\ N_{rxn},\ N_{hor}) \geq 0 \tag{4-4}$$

$$t_r^* > 0 \tag{4-5}$$

$$T_c^* > 0 \tag{4-6}$$

The final dimensionless version of the optimization problem has a nonlinear objective function, one linear inequality constraint (g_1), two nonlinear inequality constraints (g_2, g_3), and bound constraints for the optimization variables. The two remaining variables are the dimensionless reaction time (t_r^*) and column cycle time(T_c^*). The three constraints

correspond to non-negative idle times for each stage (g_1, g_2) and a minimum average production rate (g_3). Because one of the two idle time inequality constraints will always be tight at the optimum, the reaction time is really the only independent decision variable.

A total of ten user supplied dimensionless parameter groups are required. These dimensionless groups are given in Table 4-1 with a description of the appropriate ratios involved. Of the ten dimensionless groups, only one (N_{rxn}) affects both the objective function and the constraints. N_{rxn} is the key dimensionless group because the primary performance effect in this example is the conversion in the reactor. N_{rxn} directly impacts on the character of the reactor composition profiles obtained for this problem.

As shown in Figure 4-6, the inequality constraints and variable bounds can be used to construct the feasible region in the space of the dimensionless decision variables T_c^* and t_r^*. The non-negativity constraints for T_c^* and t_r^* are redundant constraints in this case. The remaining three constraints (non-negative idle times and minimum average rate) are located on the plot based on the values of the dimensionless groups N_{rate}, N_{chr}, N_{chc}, N_{hor}, and N_{rxn}. Table 4-1 also gives the values used for the dimensionless parameter groups for this problem.

As shown in Figure 4-6, the feasible region can have a finite size. However, for other parameter values, a feasible region may not exist at all. This case is illustrated in Figure 4-7 by changing the value of N_{hor} from 0.476 to 0.600. The interpretation of this case is that the required amount of B cannot be produced in the specified time using the given equipment. Relaxing the horizon time constraint slightly would be one way to make a feasible solution possible again. This example illustrates that obtaining a feasible solution to the performance subproblem may not always be possible with the given equipment and constraints.

A successive quadratic programming (SQP) algorithm from the NAG Library has been used to solve the NLP minimization problem (Numerical Algorithms Group, 1984). The optimal solution for the base case has a reaction time of 1.658 and a column cycle time of 0.945 (both scaled). These values generated a scaled objective value of 5.677. Although the feasible region for the example problem is slightly non-convex, the solution method appeared to find the global optimum. Multiple starting points were used in an attempt to locate other solutions, but none was found. The non-convexity, caused by the reaction kinetics, appears in the column idle time constraint (g_2) because the column time is dependent on the amount of A and B produced.

TABLE 4-1
DIMENSIONLESS GROUPS AND BASE CASE PARAMETER VALUES

N_{rxn}	$\frac{\text{Rate B} \rightarrow \text{C reaction}}{\text{Rate A} \rightarrow \text{B reaction}}$	0.500
N_{cap}	$\frac{\text{Rental charges}}{\text{Feed costs}}$	1.067
N_C	$\frac{\text{Byproduct costs}}{\text{Feed costs}}$	0.400
N_A	$\frac{\text{Reactant recycle costs}}{\text{Feed costs}}$	0.200
N_{clean}	$\frac{\text{Cleaning costs}}{\text{Feed costs}}$	0.313
N_u	$\frac{\text{Utility costs}}{\text{Feed costs}}$	0.450
N_{chr}	$\frac{\text{Reactor changeover time}}{\text{Characteristic reaction time}}$	0.502
N_{chc}	$\frac{\text{Column changeover time}}{\text{Characteristic column time}}$	0.263
N_{rate}	$\frac{\text{Characteristic reactor "rate"}}{\text{Characteristic column "rate"}}$	5.330
N_{hor}	$\frac{\text{Required "rate"}}{\text{Average "rate"}}$	0.476

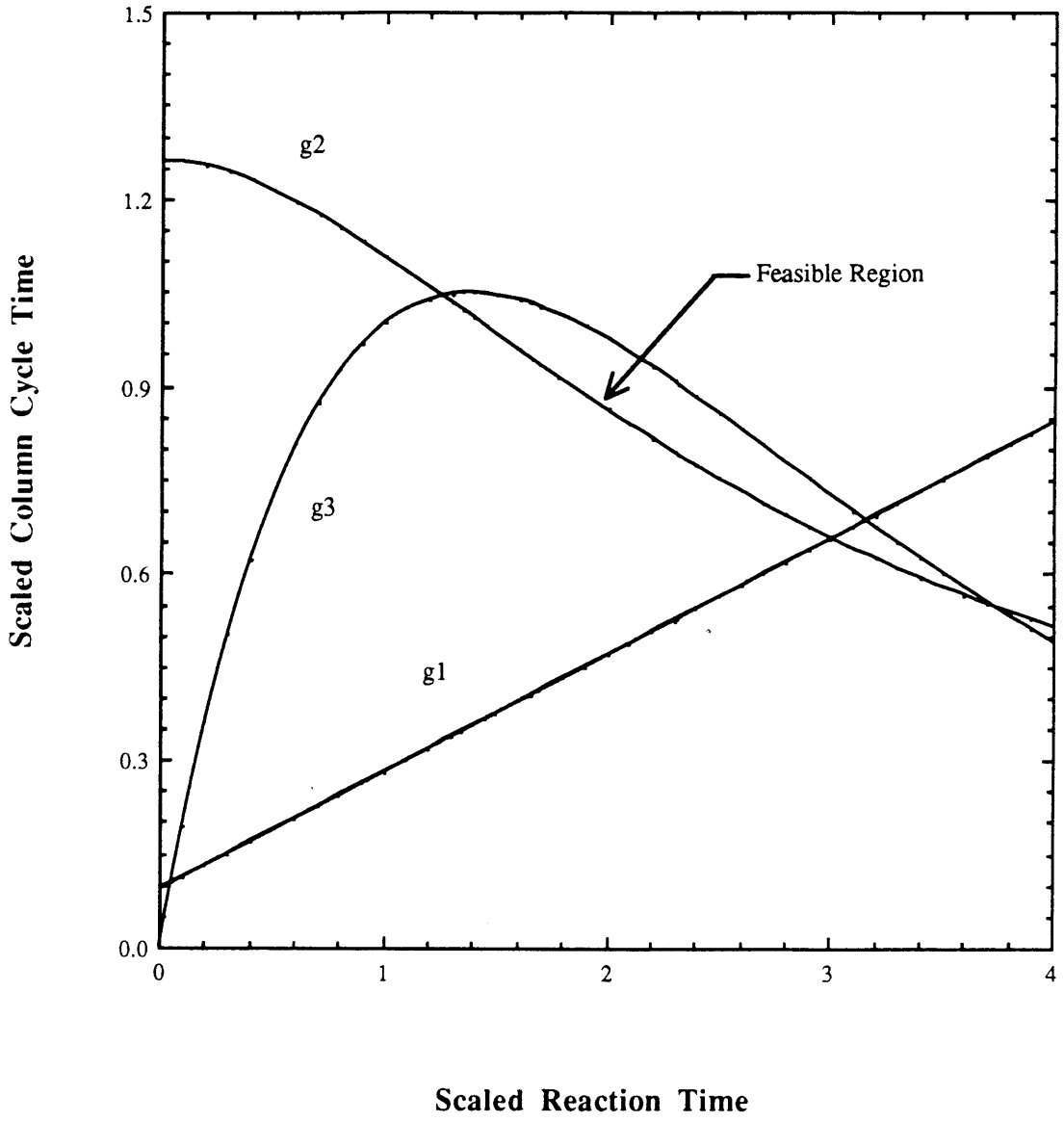


FIGURE 4-6. FEASIBLE REGION FOR BASE CASE PARAMETER VALUES FOR EXAMPLE PROBLEM #1

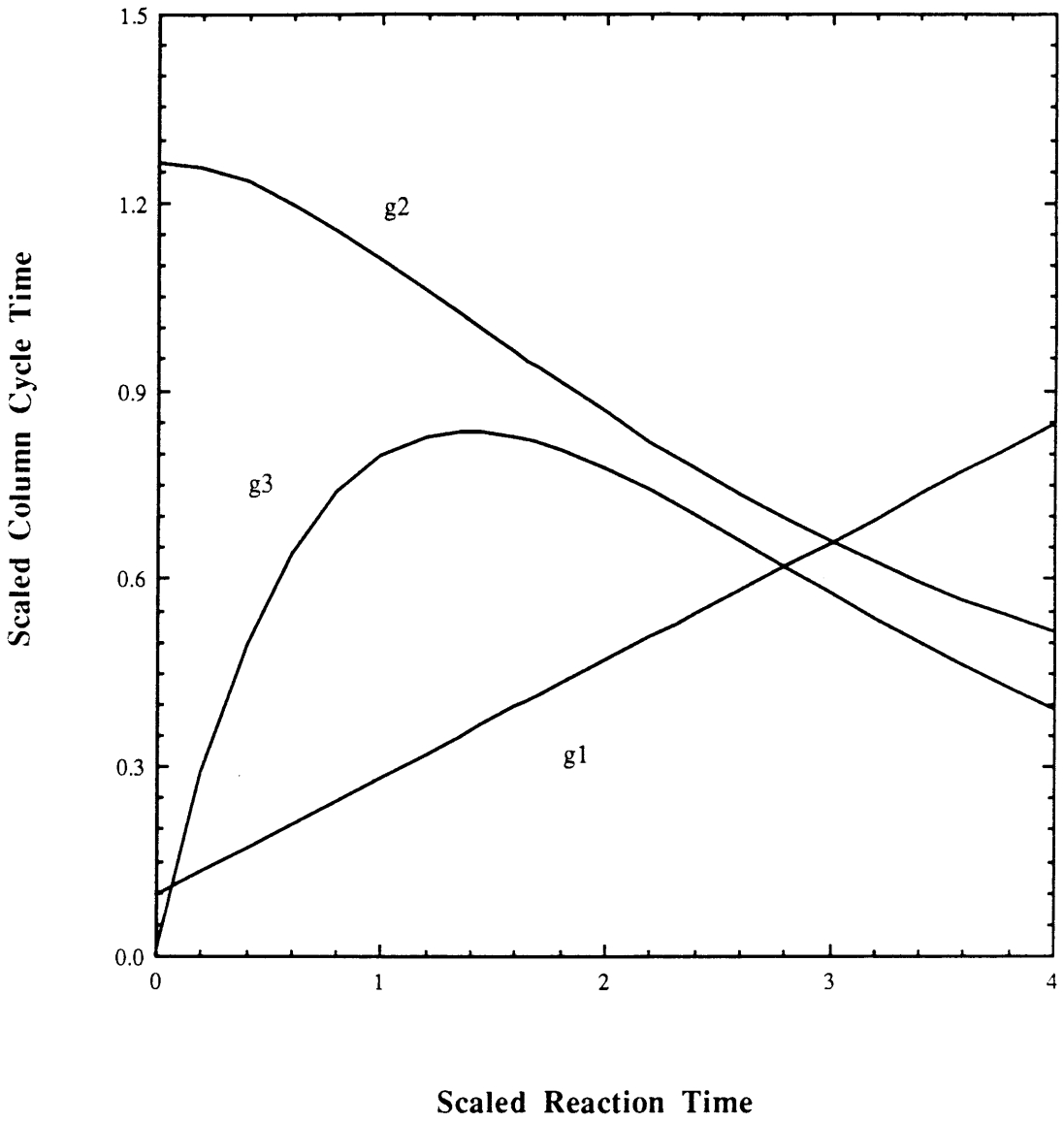


FIGURE 4-7. CASE WITHOUT FEASIBLE REGION

The global optimality of the solution obtained by the SQP can be verified graphically in this case. Figure 4-8 shows a plot of the feasible region with contours of various objective function values overlaid. The optimal solution occurs at the point where the objective function contour is just tangent to the feasible region. It can be shown for this problem that the optimal solution will always lie on one of the two idle time constraints (g_1 or g_2). Any point in the interior of the feasible region will be non-optimal. This fact is intuitively reasonable because we would expect at least one unit in the process to have no idle time in the optimal case. If both did have idle time, the two idle times could be reduced by the smaller of the two to make one idle time zero without affecting the feasibility of the solution.

A series of runs made with fixed values of the reaction time determined the corresponding optimal values of the column cycle time and the objective function. The results for the objective function, shown in Figure 4-9, demonstrate the magnitude of the potential benefits that can be obtained by including performance in the optimization. There is clearly an optimal reaction time. The overall systems approach proposed here allows performance interactions among the various units to reduce the total process costs. Figure 4-10 shows the optimal column cycle time for the various choices of reaction time. The reversal in slope of the curve occurs at the point where the optimal solution jumps from one idle time constraint to the other. For reaction times below 2.99, the column is the bottleneck unit and has no idle time. For reaction times above 2.99, the reactor becomes limiting, and the column has idle time. The column cycle time changes significantly as the reaction time is varied.

The benefits of this performance optimization approach are demonstrated by comparing the optimal solution with one obtained using engineering "common sense." The reaction kinetics cause the concentration of product B to go through a maximum. Intuitively, setting the reaction time to correspond to this maximum would minimize the number of batches needed to make the required amount of product B. In addition, this time corresponds to the maximum yield of product on reactant A. For this example, the dimensionless reaction time for maximum product concentration is 1.386. Going past this point reduces the concentration of product B and increases the concentration of byproduct C. The higher disposal cost of C compared to that for unused reactant A would seem to be an additional indication that the reaction should be terminated at 1.386.

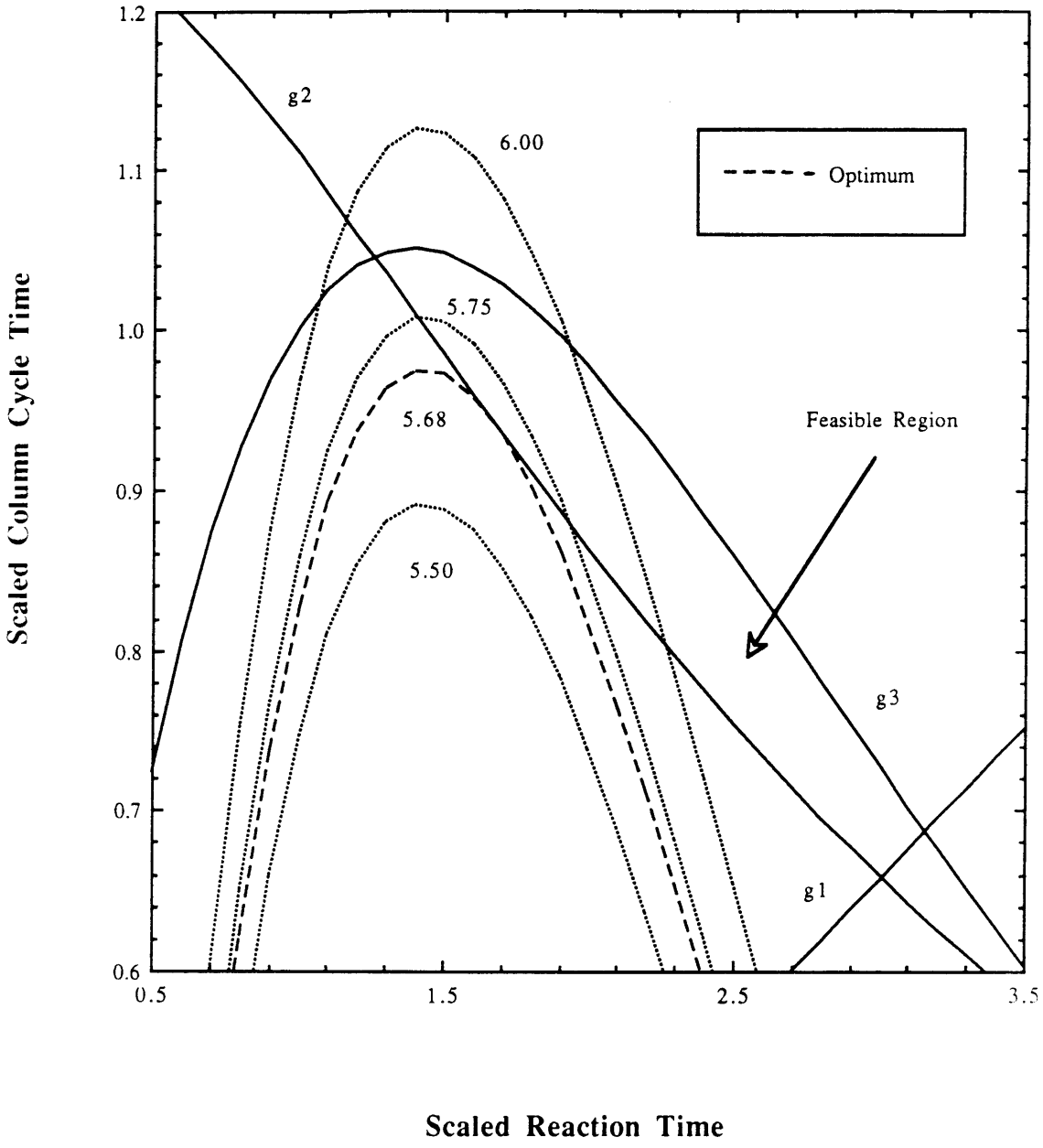


FIGURE 4-8. GRAPHICAL VERIFICATION OF OPTIMUM FOR EXAMPLE PROBLEM #1

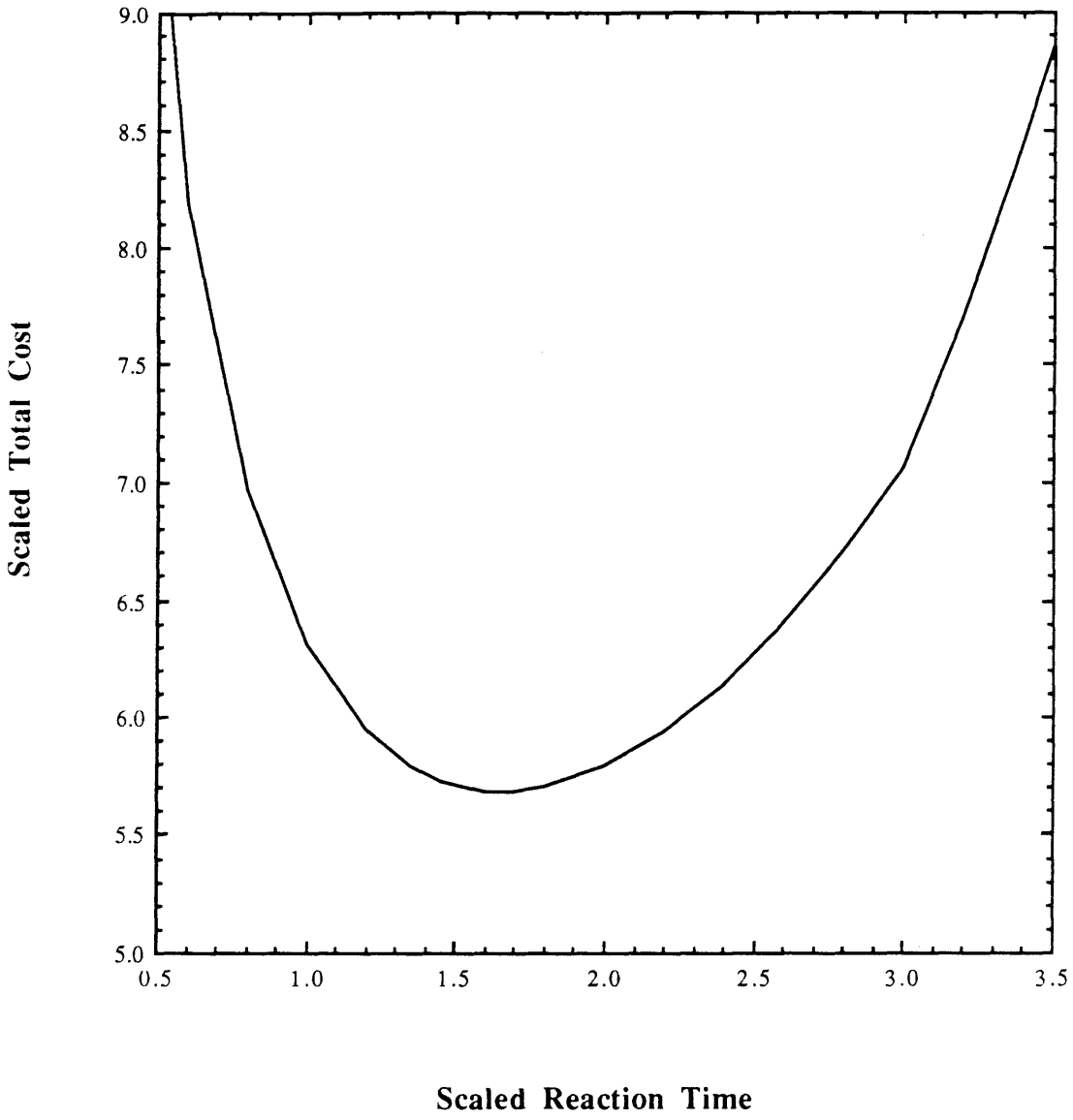


FIGURE 4-9. TOTAL COST AS A FUNCTION OF REACTION TIME FOR EXAMPLE PROBLEM #1

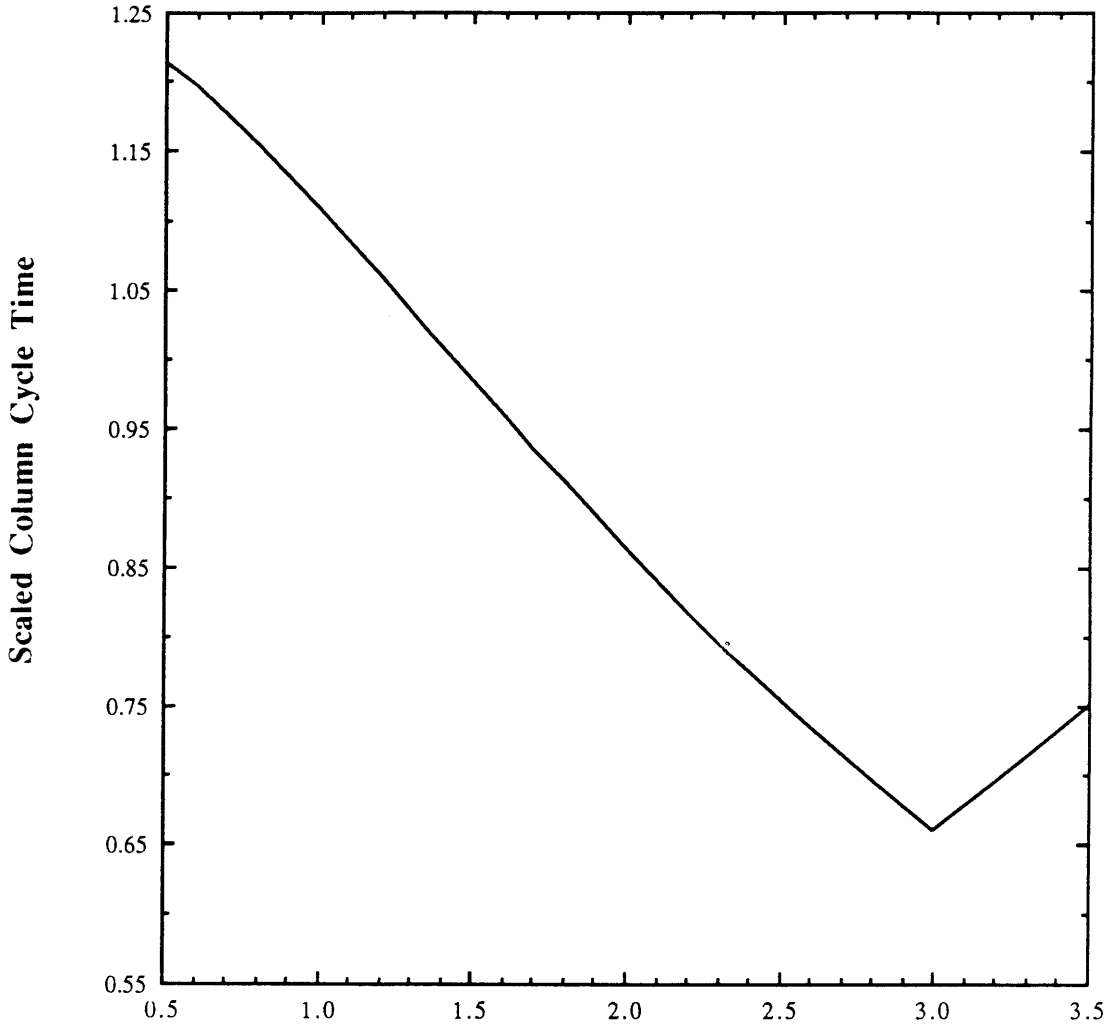


FIGURE 4-10. COLUMN CYCLE TIME AS A FUNCTION OF REACTION TIME FOR EXAMPLE PROBLEM #1

However, the optimal dimensionless reaction time has already been shown to be 1.658. Table 4-2 summarizes the results for the two cases. The reactor and column operation times are significantly different (19.5%, 6.6%, respectively) for the two cases, but the change in the total manufacturing cost is small (1.6%). The longer reaction time in the optimal case results in an easier separation. By allowing the reaction to proceed longer, the amount of A and B to be taken overhead by the column is reduced, resulting in a shorter separation time in the column. Since the column is the bottleneck unit, reducing the column cycle time shortens the time required for the optimal case to finish the campaign. The lowered equipment rental costs outweigh the increases in raw materials costs. This example problem clearly shows the trade-off obtained by shifting the performance load between the two stages..

This example problem also illustrates the trade-off between processing efficiency and processing speed. In this two stage problem, this trade-off is obtained by a performance load shift. A series of optimizations have been done with a range of total equipment rental charge rates. Naturally, the overall cost increases as the rental costs increase. What is interesting are the shifts in the optimal reaction time and campaign time.

These shifts are shown in Figure 4-11, in which the optimal reaction time and resulting campaign time are plotted as a function of the magnitude of the sum of the rental charge rates. As the cost of the equipment increases, the campaign time is reduced by increasing the reaction time. The longer reaction time reduces the time requirement for the second stage, which is the bottleneck. The increased processing rate comes at the expense of less efficient processing, which results in increased costs for raw materials, waste treatment, and utilities. At the limit of infinite rental rates, the objective becomes minimization of the campaign time. Clearly then, the best processing conditions depend on the magnitudes of the equipment usage charges, and hence, the overall plant environment.

The value of N_{rxn} also has a significant effect on the optimal solution. Changing N_{rxn} amounts to shifting the relative rates of the two reactions. Increasing N_{rxn} increases the relative rate at which desirable product B is converted into byproduct C. Thus, lowering N_{rxn} is expected to produce a lower cost solution. Figure 4-12 shows the cost as a function of reaction time for three values of N_{rxn}. Table 4-3 summarizes the optimal results for the three cases. The optimal cost decreases, the optimal reaction time increases, and the curvature in the vicinity of the optimum flattens out as N_{rxn} is reduced. For higher

TABLE 4-2.
COMPARISON OF OPTIMAL AND MAXIMUM CONVERSION CASES

	Optimal Solution	Maximum Conversion Solution
Scaled reaction time, t_r^*	1.658	1.368
Scaled column cycle time, T_C^*	0.945	1.013
Mole fraction of A in column feed, x_A	0.191	0.250
Mole fraction of B in column feed, x_B	0.492	0.500
Mole fraction of C in column feed, x_C	0.317	0.250
Scaled campaign time, T_{tot}^*	0.915	0.964
Scaled cost components:		
Raw materials	2.033	2.000
Waste Treatment, Clean-outs	0.970	0.924
Column Utilites	0.625	0.675
Equipment Usage	2.049	2.161
Total operating costs	5.677	5.760

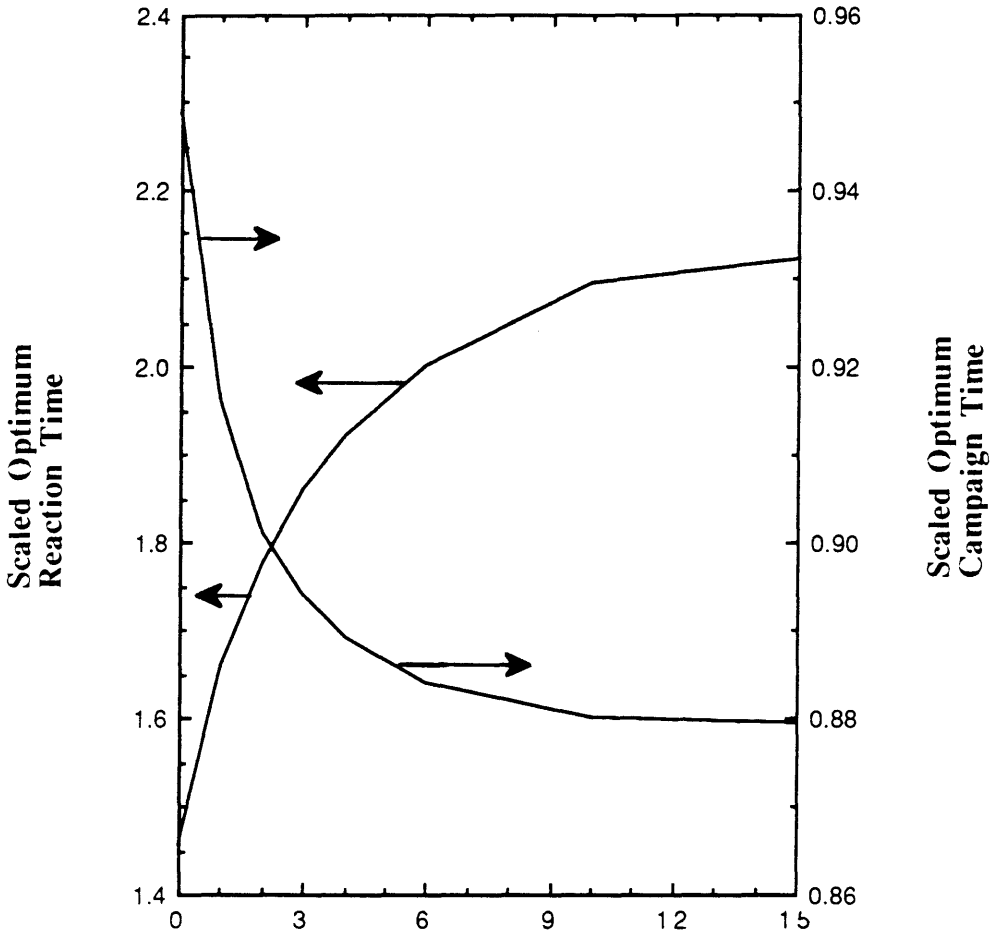


FIGURE 4-11. EFFECT OF PLANT ENVIRONMENT ON OPTIMAL DESIGN

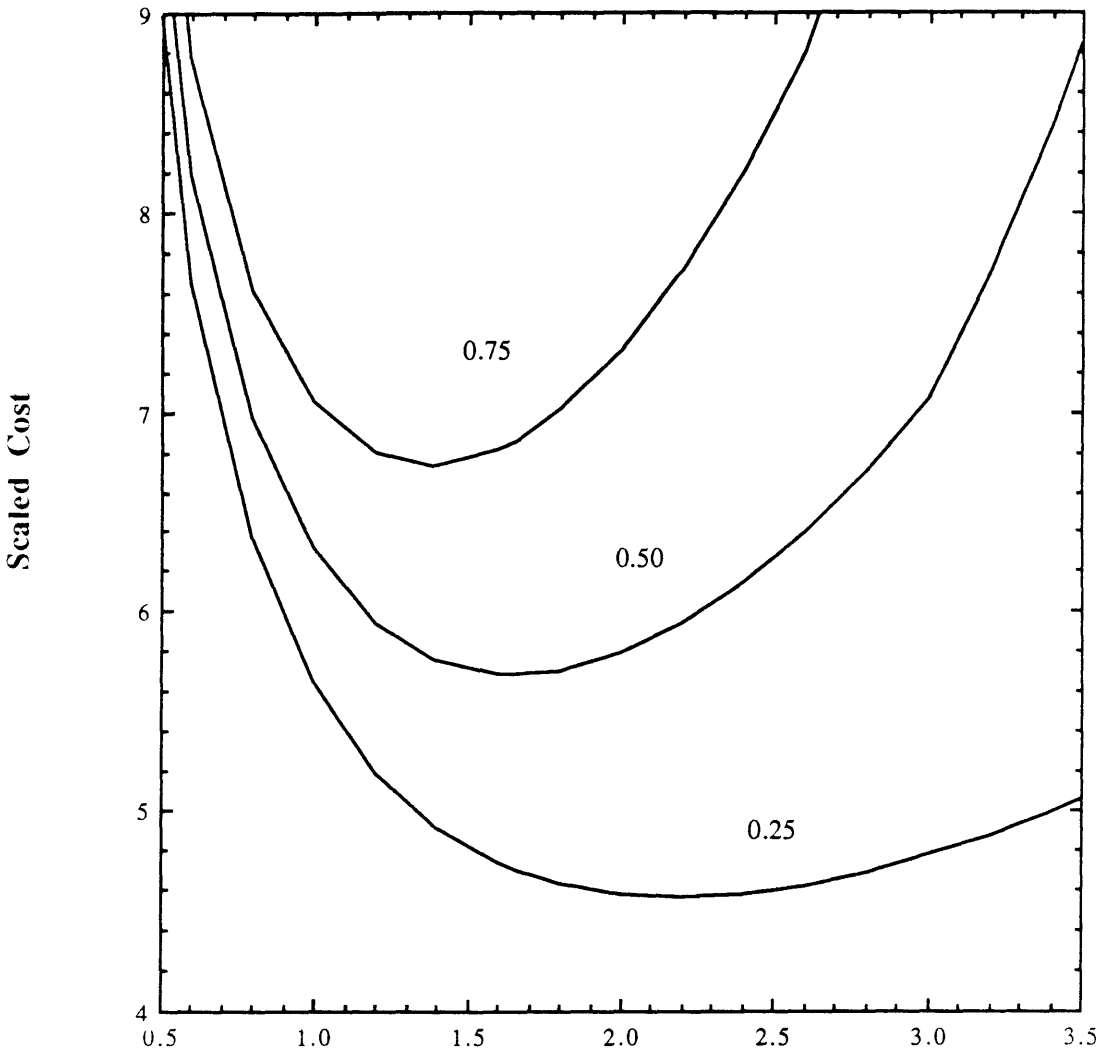


FIGURE 4-12. EFFECTS OF N_{rxn} ON TOTAL COST CURVES FOR EXAMPLE PROBLEM #1

TABLE 4-3.
EFFECT OF VARYING N_{rxn} ON OPTIMAL SOLUTION

	N_{rxn} 0.25	N_{rxn} 0.50	N_{rxn} 0.75
Optimal Reaction Time	2.187	1.658	1.380
Optimal Column Cycle Time	0.997	0.945	0.929
Mole fraction of A in column feed, x_A	0.112	0.191	0.252
Mole fraction of B in column feed, x_B	0.622	0.492	0.415
Mole fraction of C in column feed, x_C	0.266	0.317	0.333
Scaled Total Campaign Time	0.763	0.914	1.066
Scaled Total Cost	4.557	5.677	6.722

values of N_{rxn}, the penalty for choosing a non-optimal reaction time is more severe than for lower values.

Example Problem #1 has demonstrated a performance load trade-off for a simple two stage process. The effect of the overall plant environment on the optimal processing conditions through the setting of the rental charges has also been shown. These equipment usage charges allow the processing efficiency versus processing speed trade-off to be explored quantitatively. This example problem was also small enough that a two dimensional representation of the feasible region could be used to illustrate the issues of convexity and feasibility.

4.5.2 Example Problem #2

The second example problem demonstrates a processing intensity versus cycle time trade-off, which represents the second type of performance trade-off discussed earlier. The first example problem did not have any operating variables that could be manipulated, so only a performance load trade-off existed. By including the temperature of the reactor as an operating variable to be optimized, a processing intensity versus cycle time trade-off can be illustrated. After the outlet concentrations of the reactor are specified, various reaction temperatures and times are tried in order to minimize the cost of the process.

The same basic two stage process from Example Problem #1 is used for this second example problem. However, this problem incorporates temperature dependent reaction kinetics. Arrhenius expressions are included to relate the reaction rate constants to the reaction temperature. In order to prevent a change in the relative rates of the two reactions, the activation energies are assumed to be equal. This restriction means that raising the temperature only accelerates the rate. A utility cost term for the reactor is added to the objective function to reflect cost differences caused by running the reactor at different temperatures. Appendix A.2 describes in detail the model equations for Example Problem #2.

The objective of the optimization is the minimization of total operating costs. This cost function includes terms for equipment rental, raw materials, waste treatment, utilities, and unit clean-outs. The optimization problem consists of minimizing a nonlinear objective function, subject to eight equality constraints, five inequality constraints, and bounds on the eleven variables. Again, the equality constraints can be used to reduce the number of

optimization variables to three (reaction time, reaction temperature, and column cycle time). Values for the twenty-four problem parameters are given in Appendix A.2.

In order to show only the processing intensity versus cycle time trade-off, the final composition in the reactor is specified. This additional constraint means that the reaction time and temperature are no longer independent. Since one of the two idle time constraints will always be tight, there is really only one degree of freedom. By varying the reaction time, the trade-off in total cost can then be illustrated.

Figure 4-13 shows the reactor temperature required to give a final mole fraction of 0.49998 for product B as a function of reaction time. As expected, the temperature decreases as the reaction proceeds for longer reaction times because lower reaction rates are required. Increasing the reaction time also decreases the average production rate for the reactor. Because the final reactor composition is fixed, the column operating with perfect splits always takes the same amount of time to finish a batch. Since the column size is given, the average production rate of the column is constant. Thus, the choice of reaction time directly determines the location of the bottleneck stage and the total cost.

Figure 4-14 shows the total cost as a function of reaction time for the two stage process with unlimited intermediate storage. For reaction times below 38.1 hours, the column is the bottleneck stage. As long as the column is the bottleneck stage, increasing the reaction time lowers the total cost by reducing the reactor temperature (and thus the utility costs). Clearly, there is no benefit to using the more intense processing conditions (higher temperature) to reach the specified conversion sooner than necessary.

However, once the reactor becomes the bottleneck unit, further increases in the reaction time lengthen the campaign time. Longer campaign times result in higher costs for equipment usage. These increases in reaction time will only be beneficial if the savings in utility costs outweigh the additional equipment costs. For this example, the equipment costs are larger, so the optimal reaction time occurs when the average production rates of the two stages are equal. Thus, neither stage has any idle time. Had the savings in utility costs outweighed the increased equipment usage charges, then a longer reaction time would be optimal. In such a case, the column would have idle time at the optimal solution.

Figure 4-15 shows the cost trade-off for the no intermediate storage case. When operating with no intermediate storage, the two stages must have the same batch sizes and cycle times. Although the reactor is twice as large as the still for the column in this

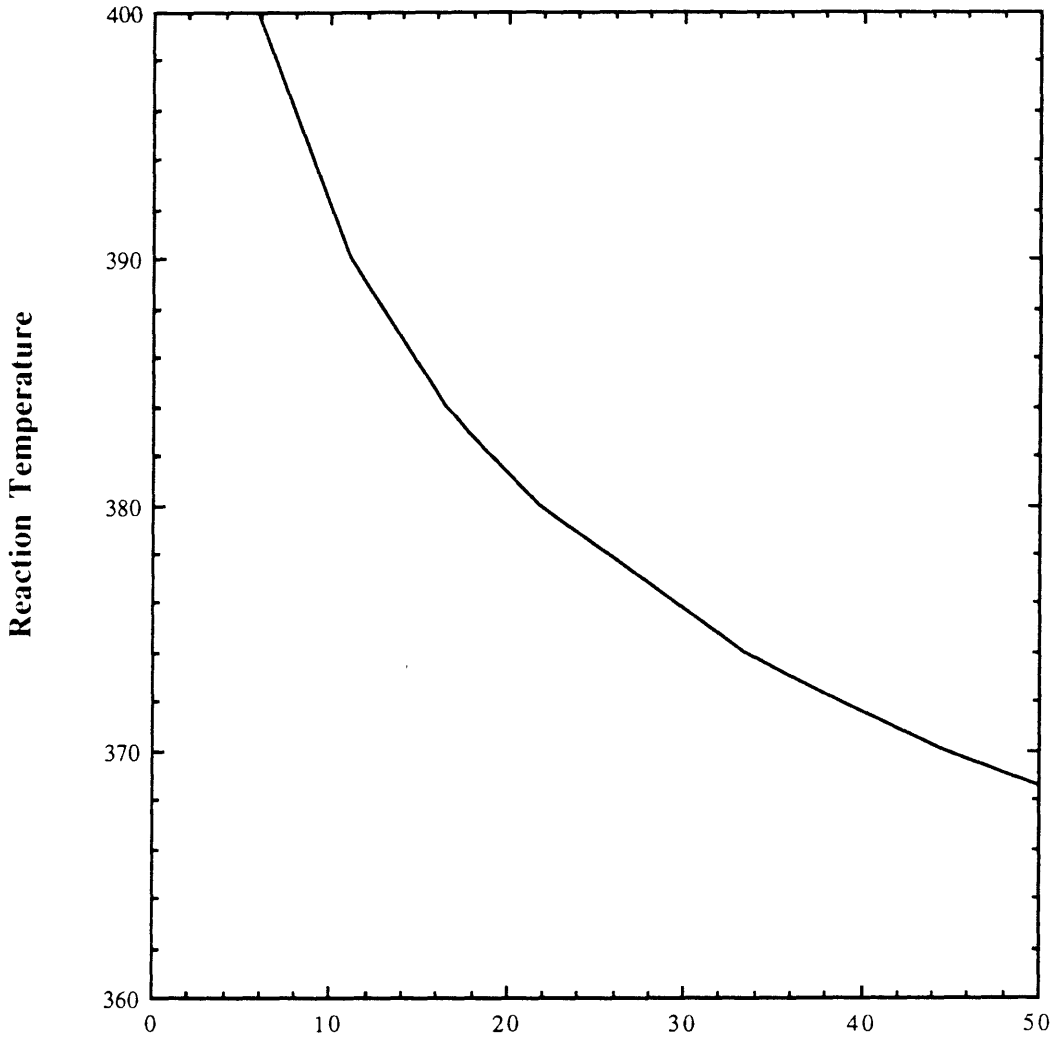


FIGURE 4-13. LOCUS OF REACTION TIMES AND TEMPERATURES RESULTING IN x_B=0.500 FOR EXAMPLE PROBLEM #2

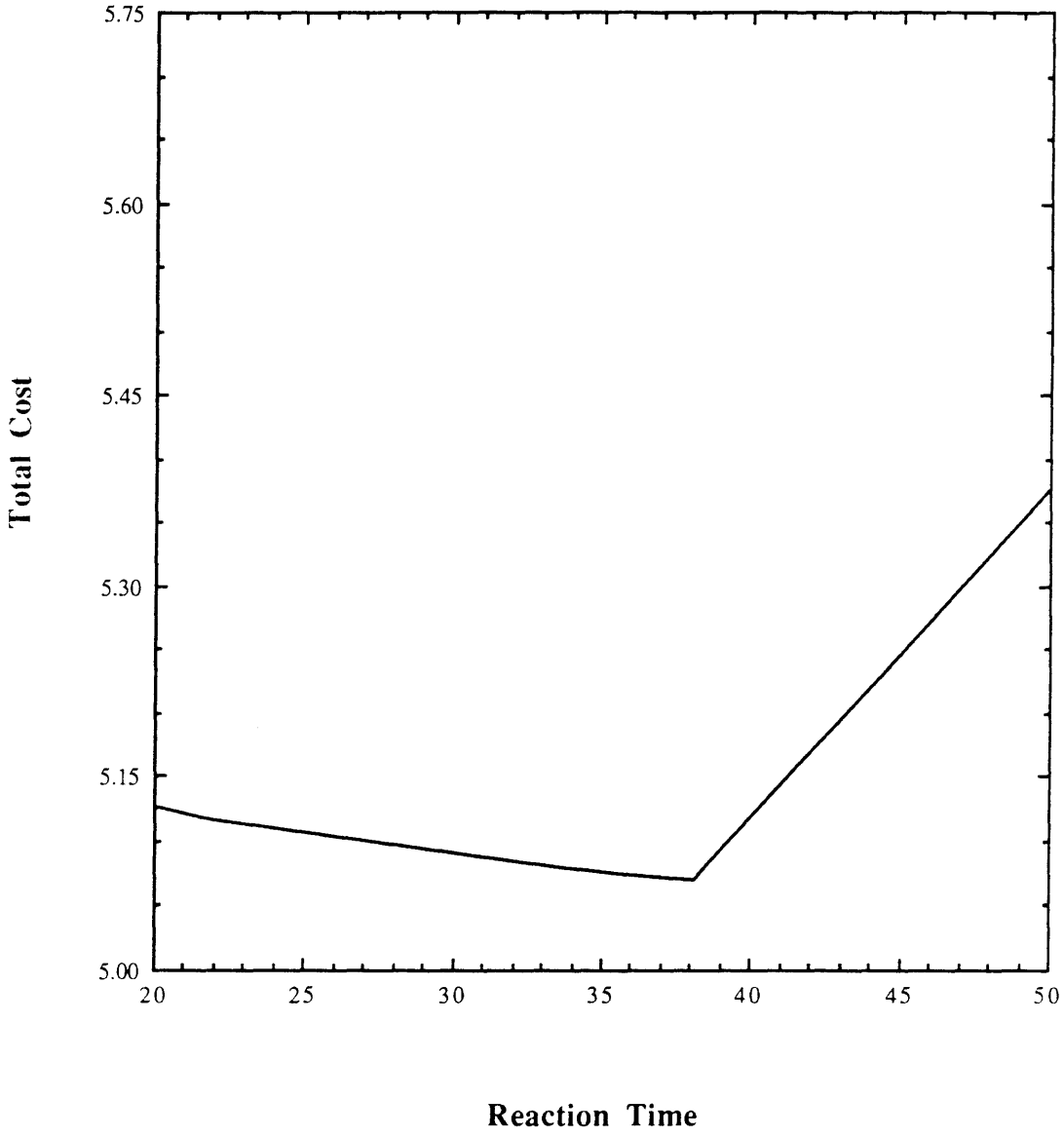


FIGURE 4-14. TOTAL COST AS A FUNCTION OF REACTION TIME FOR UIS FOR EXAMPLE PROBLEM #2

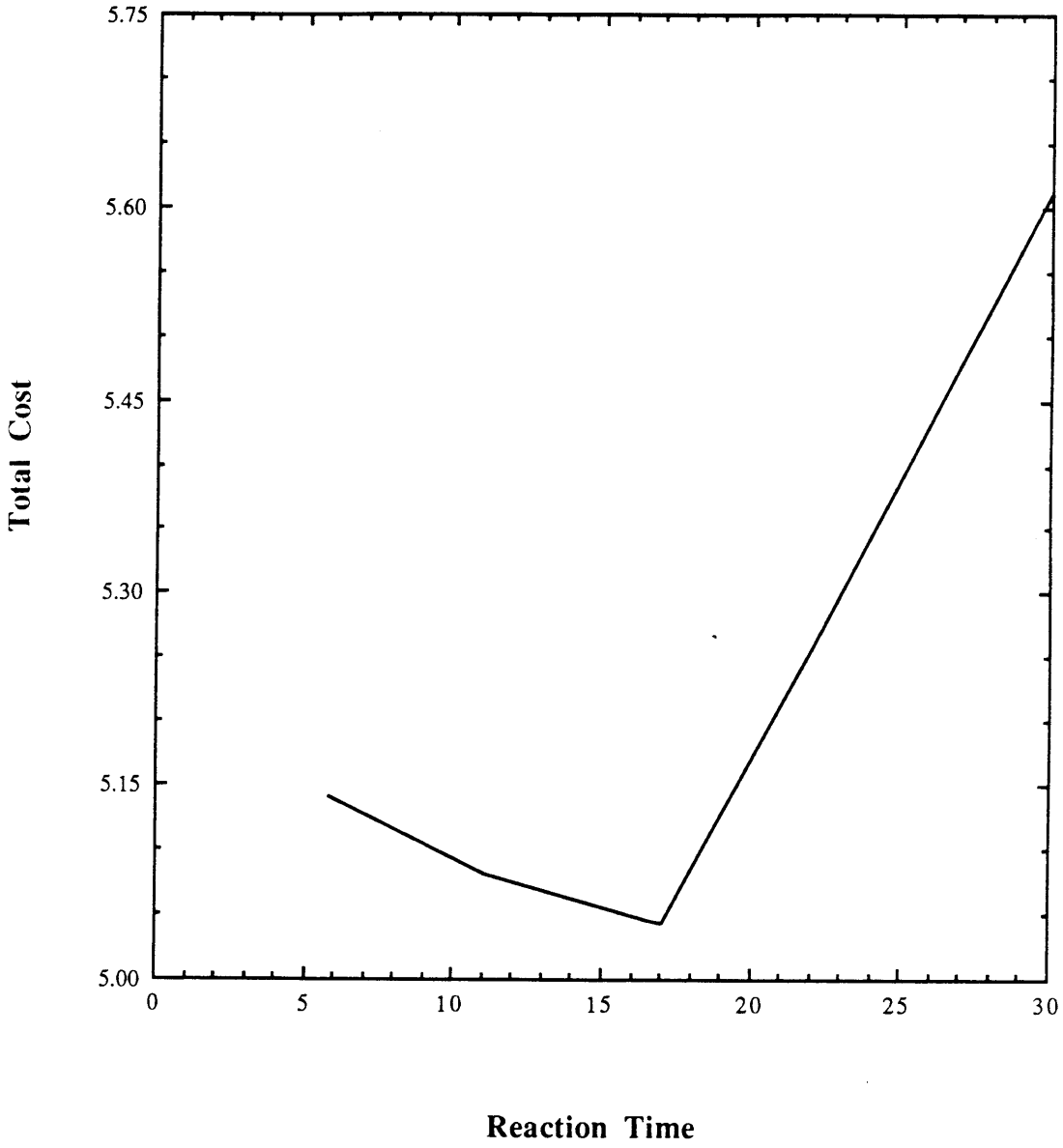


FIGURE 4-15. TOTAL COST AS A FUNCTION OF REACTION TIME FOR NIS FOR EXAMPLE PROBLEM #2

example, the equal batch size constraint means that the reactor cannot operate full. Thus, the average production rates for the NIS case are equal at a shorter reaction time than the UIS case described above. The NIS case is slightly less costly (even though the reactor runs only half full) because there is no usage cost for intermediate storage.

The choice of 0.49998 as the mole fraction of product B leaving the reactor has been made arbitrarily. However, the maximum mole fraction possible for product B is 0.500 for these kinetics. Based on experience gained from Example Problem #1, the results obtained by exploring the processing intensity versus cycle time trade-off for this high product mole fraction are expected to be reasonably close to optimal. By allowing the product mole fraction to vary, a performance load trade-off is introduced. An overall optimum can be determined by exploring both types of trade-offs simultaneously.

Table 4-4 summarizes the results of the NLP optimizations for cases when both trade-offs are considered and when only the processing intensity trade-off is allowed. Slight improvements in the cost are obtained by including the performance load trade-off. For both UIS and NIS cases, the performance load shifts caused the reactions to go slightly further. Since less A and B are present at the end of the reaction in these cases, the column operation times are reduced enough to increase the average rates. The temperatures of the reactors are also increased slightly so that they reach their new endpoints sooner. Overall, higher bottleneck average production rates result in shorter campaign times. These changes cause increases in the costs of raw materials, waste treatment, reactor utilities, and unit clean-outs. However, these increases are offset by lower column utility and equipment usage costs.

Example Problem #2 has demonstrated a processing intensity versus cycle time trade-off for a simple two stage process. Effects of the intermediate storage policy on the optimal values for operating conditions have also been shown. Finally, by relaxing a restrictive assumption in the example problem, both types of trade-offs were explored simultaneously by the optimization method to determine an overall optimal solution.

4.5.3 Example Problem #3

The effects of model complexity on the performance of the solution method are studied in the third example problem. In particular, solution accuracy and computational load are examined. Although the same basic two stage process from Example Problem #2 is used here, the values of some problem input parameters have been changed. The main

TABLE 4-4.
COMPARISON OF OPTIMAL RESULTS FOR EXAMPLE PROBLEM #2

	PROCESSING INTENSITY TRADE-OFF ALONE		BOTH TYPES OF TRADE-OFFS	
Constraint on Conversion	YES	YES	NO	NO
Storage Policy	NIS	UIS	NIS	UIS
Optimal Results				
Reaction Time	17.05	38.1	16.3	36.4
Temperature	383.6	372.2	385.7	374.4
Column Cycle Time	21.1	21.1	20.3	20.2
x_A	0.253	0.253	0.219	0.214
x_B	0.500	0.500	0.498	0.497
x_C	0.247	0.247	0.283	0.289
Campaign Time	87.8	87.8	85.1	84.8
Scaled Cost Results				
Raw Materials	2.000	2.000	2.008	2.011
Waste Treatment / Clean-out	0.523	0.461	0.552	0.493
Utilities	1.519	1.450	1.488	1.415
Equipment Usage	1.001	1.158	0.970	1.118
Total Cost	5.042	5.069	5.018	5.038

adjustment is that the activation energies are no longer equal. Thus, the reaction temperature affects both the absolute and relative magnitudes of the two reaction rates. Lowering the temperature increases the maximum conversion because E_1 is less than E_2. Details on the model equations are given in Appendix A.2, and problem parameter values for Example Problem #3 are provided in Appendix A.3.

The objective of the optimization is the minimization of total operating costs, including terms for equipment rental, raw materials, waste treatment, utilities, and unit clean-outs. Both types of performance trade-offs are explored in the optimization of this problem. Optimizations are done for a series of reaction times to generate the "optimal" cost and reaction temperature. Figure 4-16 illustrates the combined effects of the two types of performance trade-offs for the UIS case. Clearly, there are potential cost benefits to be obtained by solving the performance subproblem. The best reaction temperature also changes significantly depending on the choice of reaction time.

Table 4-5 summarizes the optimal solutions for the NIS and UIS cases. As observed previously in Example Problem #2, the NIS case has a shorter optimal reaction time and a higher optimal reaction temperature than the UIS case in order to mitigate the effects of running the reactor only half full. The UIS case is less costly in this example in almost all cost areas with the exception of equipment usage because of the storage costs.

The process model for the two stage process uses analytic expressions to relate the final component mole fractions to the reaction time and reaction temperature. In most cases, kinetic models for batch reactions will not be easily integrated to obtain closed form expressions. In these cases, the ordinary differential equations must be integrated numerically to obtain final reaction compositions.

Numerical integration of ordinary differential equations greatly increases the computational effort required to model batch processes. Also, error incorporated during integration might affect the performance of the optimization routine. Integration error can be reduced by using smaller time steps or higher order methods, but both these remedies increase the computational effort. Thus, the choices for integration method and time step introduce a trade-off between the accuracy and computational load of solution procedures for the performance subproblem.

The possible effects of numerical integration are demonstrated by replacing the analytic model for the reaction kinetics with differential equations for the mole fractions of

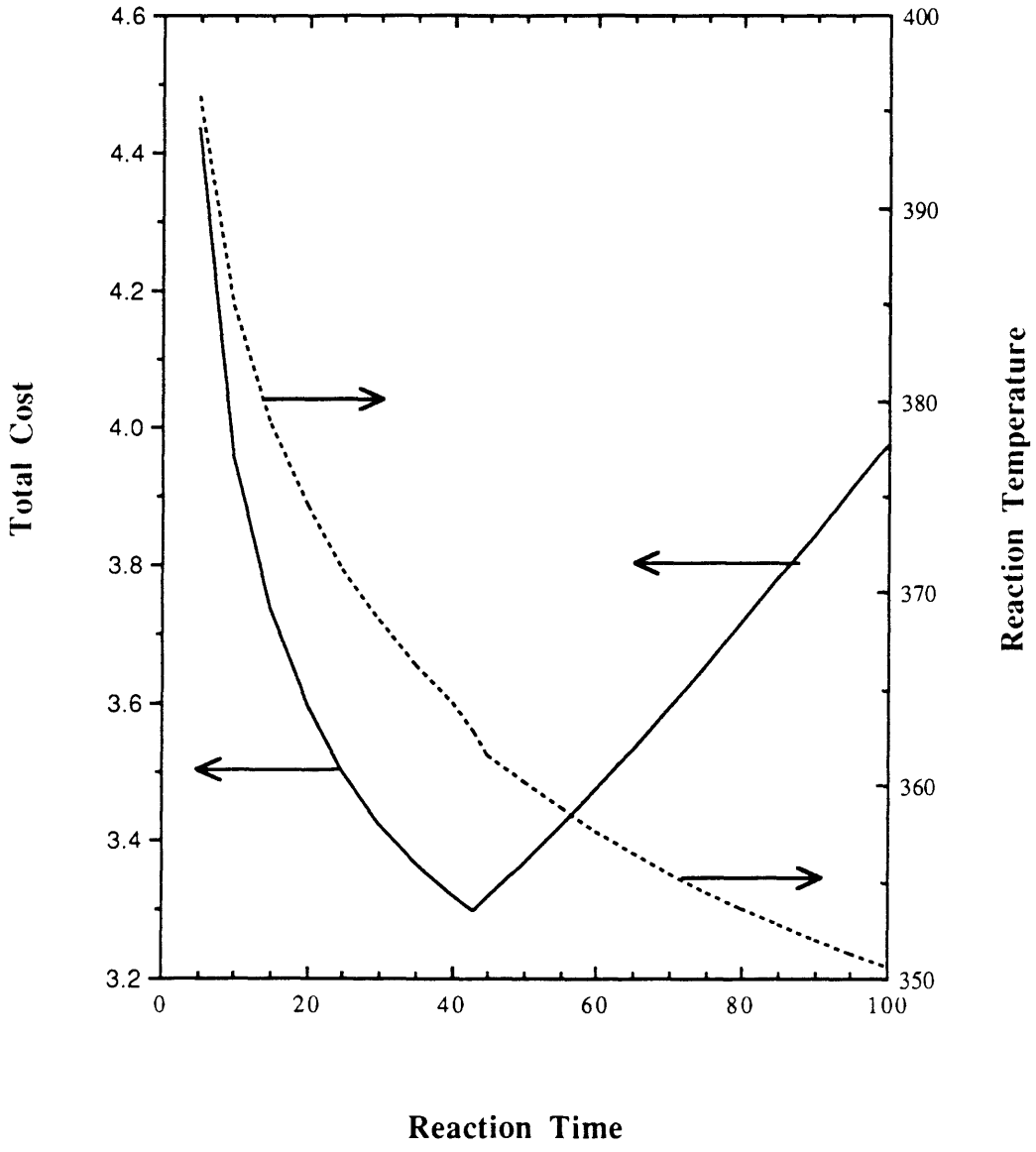


FIGURE 4-16. OPTIMAL TOTAL COST AND REACTION TEMPERATURE AS A FUNCTION OF REACTION TIME FOR EXAMPLE PROBLEM #3

TABLE 4-5.
COMPARISON OF OPTIMAL SOLUTION FOR UIS AND NIS POLICIES FOR EXAMPLE PROBLEM #3 (PERFECT SPLITS COLUMN)

	NIS	UIS
Reaction Time	19.1	42.9
Temperature	374.2	362.8
Column Cycle Time	23.1	23.5
x_A	0.137	0.094
x_B	0.717	0.779
x_C	0.146	0.127
Campaign Time	67.1	62.7
Scaled Cost Results:		
Raw Materials	1.394	1.283
Waste Treatment / Clean-out	0.270	0.188
Utilities	1.115	0.997
Equipment Usage	0.765	0.828
Total Cost	3.544	3.296

A and B. An explicit first order Euler integration method with fixed step size is used to numerically integrate the differential equations to obtain the compositions in the reactor as a function of time. The constraint that all mole fractions sum to unity is imposed by calculating the mole fraction of C at each time step using the current mole fractions for A and B.

The analytic solution shown above in Table 4-5 serves as the point of comparison for the optimizations with the numerically integrated kinetics expressions. Using a number of different fixed step sizes shows the effects of model error on the performance of the optimization method. The same starting points have been used for all cases. In addition, the computational loads for the analytic and numerical integration cases are compared.

The results for the optimizations with the different step sizes and the analytic case are shown in Table 4-6. Smaller step sizes do better at converging to the exact optimum, but require considerably more computation time. Increasing the step size decreases the computational load, although some accuracy is sacrificed. Eventually, the integration error introduced by the larger step sizes causes the calculations to go unstable. These results illustrate the general trade-off between calculational accuracy and computational load which must be considered whenever numerical integration of differential equations must be done.

In addition to requiring the integration of differential equations, the element of time can increase the complexity of operation for batch processes. For example, reaction temperature need not be constant with time, and a time-varying temperature profile might improve process performance. More detailed process models would be required to evaluate these types of more complicated operating strategies. Although the determination of an optimal profile involves solving a variational optimization problem, piece-wise constant profiles can be used to side-step this type of optimization problem. Sundaram and Evans (1989) have illustrated the use of piece-wise constant reflux ratio profiles in the optimization of batch distillation column performance. However, the number of optimization variables in the performance subproblem is increased, likely resulting in a higher computational load. In general, the use of more complicated operating strategies to improve process performance requires the solution of a more difficult performance subproblem. Thus, there is a trade-off between the quality of the final solution and the amount of effort required to obtain that level of quality.

This trade-off can be illustrated by comparing the use of the perfect splits and Sundaram column models in the optimization of the two stage problem. The Sundaram

TABLE 4-6.
EFFECTS OF NUMERICAL INTEGRATION OF DIFFERENTIAL EQUATIONS ON OPTIMIZATION OF EXAMPLE PROBLEM #3 (PERFECT SPLITS COLUMN)

Reaction Time Step Size (hr)	"Optimal" Cost	Average % Error in Decision Variables	Number of Simulations	CPU Time* (sec)
Analytic	3.2963	0.00%	120	4.16
0.001	3.2963	0.00%	120	763.45
0.01	3.2961	0.00%	120	79.99
0.05	3.2950	0.00%	88	16.42
0.25	3.2895	0.02%	80	5.66
0.5	3.2826	0.04%	80	4.54
1.0	3.2690	0.07%	84	4.27
2.0	3.2424	0.11%	84	4.05
3.0	3.2160	0.13%	80	3.92
4.0	3.3176	51.07%	260	6.77
5.0	3.1652	0.36%	72	3.54
6.0	1.4026	60.16%	400	11.95

* CPU time on a DEC Microvax II. All optimizations use the same starting point.

column model (Sundaram and Evans, 1990) uses a series of short-cut equations to relate the composition of the distillate at any instant in time to the pot composition, the reflux ratio, the number of stages, and the relativity volatilities of the components. The pot compositions are updated by integrating differential equations numerically. Thus, this model should more accurately represent actual column performance, although at considerably higher computational expense than the perfect splits model.

The two stage process has been optimized for a series of specified reflux ratios in order to compare the two column models. A purity specification of 90 percent B is specified for the product stream from the column. For the perfect splits case, increasing the reflux ratio does not affect the quality of the split but does cause the column to take longer to complete the separation. As shown in Figure 4-17, increasing the reflux ratio causes the optimal values of the reaction time and reaction temperature to change in an attempt to shift the load away from the column. This performance load shift is shown in Figure 4-18, in which the mole fraction of product B leaving the reactor is plotted against the specified reflux ratio. Increasing the composition of B reduces the number of batches required to complete the campaign because the yield per batch is increased.

Figure 4-19 shows a comparison of the optimal total costs for the perfect splits and Sundaram models for a series of fixed reflux ratios. As expected, the two models give similar results for high reflux ratios. The sharp splits obtained with high reflux ratios approximate the perfect splits assumed by the simplified model. However, the two models diverge for lower reflux ratios when the perfect splits assumption fails to match the actual column performance. In order to satisfy the purity specification, less product can be collected on each batch for the lower reflux ratios. Some product is lost, and more batches must be produced to generate the required campaign amount. These losses drive up the overall costs despite the fact that the column uses less energy and requires less time for each batch.

Using the more complex column model allows trade-offs involving the reflux ratio to be accurately represented. The better model also permits an optimal reflux ratio to be obtained. These benefits are obtained at the expense of a significantly higher computational load. Solution times for the performance subproblem increase by a factor ranging from 100 to 1000 when the Sundaram column model is used instead of the perfect splits model. Part of this big change is due to the longer computation time for simulating the process with the more complex model. The remaining increase is due to the higher number of function calls and derivative evaluations by the SQP method during optimization.

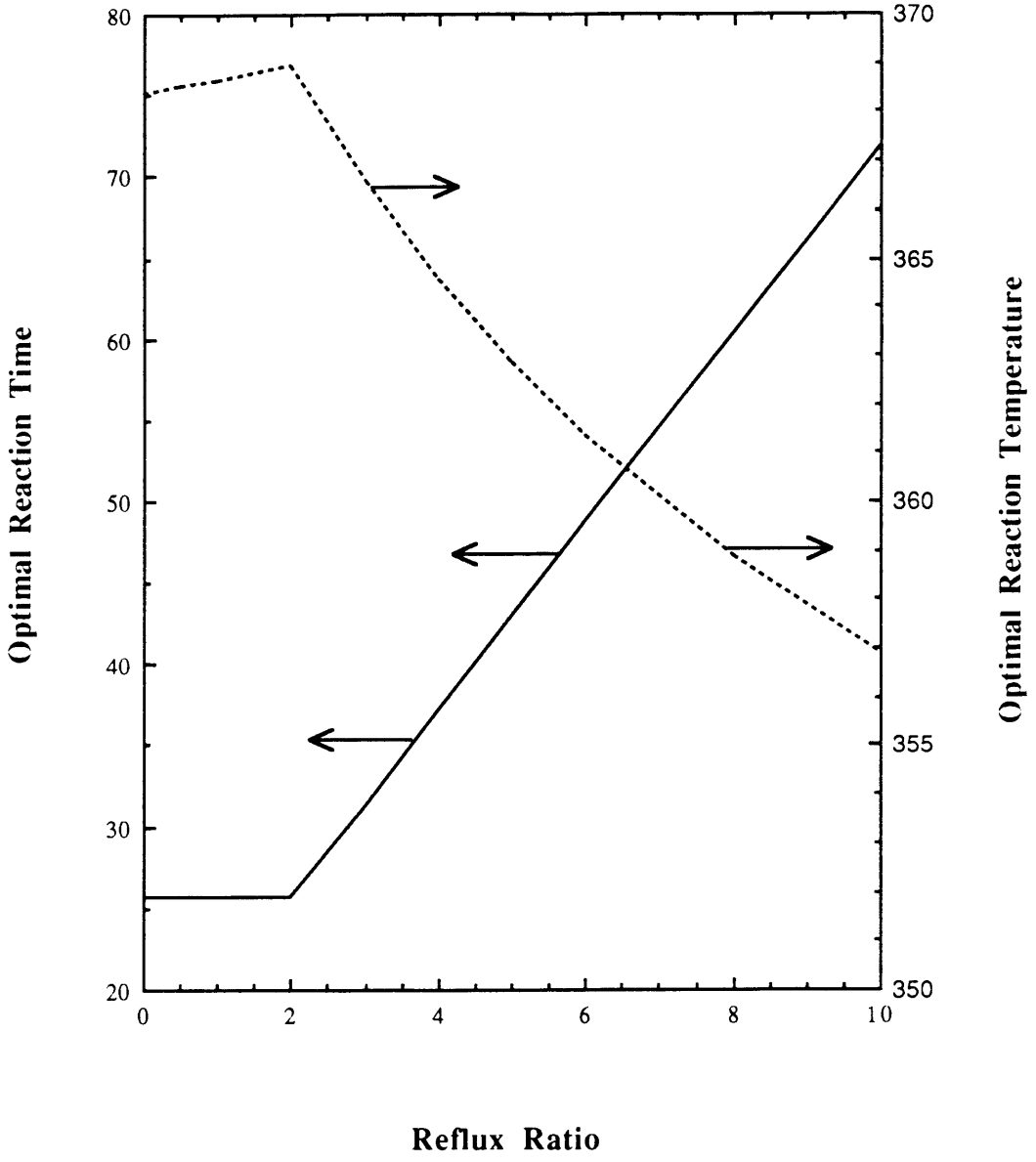


FIGURE 4-17. OPTIMAL REACTION TEMPERATURE AND TIME AS A FUNCTION OF REFLUX RATIO FOR EXAMPLE PROBLEM #3 FOR PERFECT SPLITS COLUMN

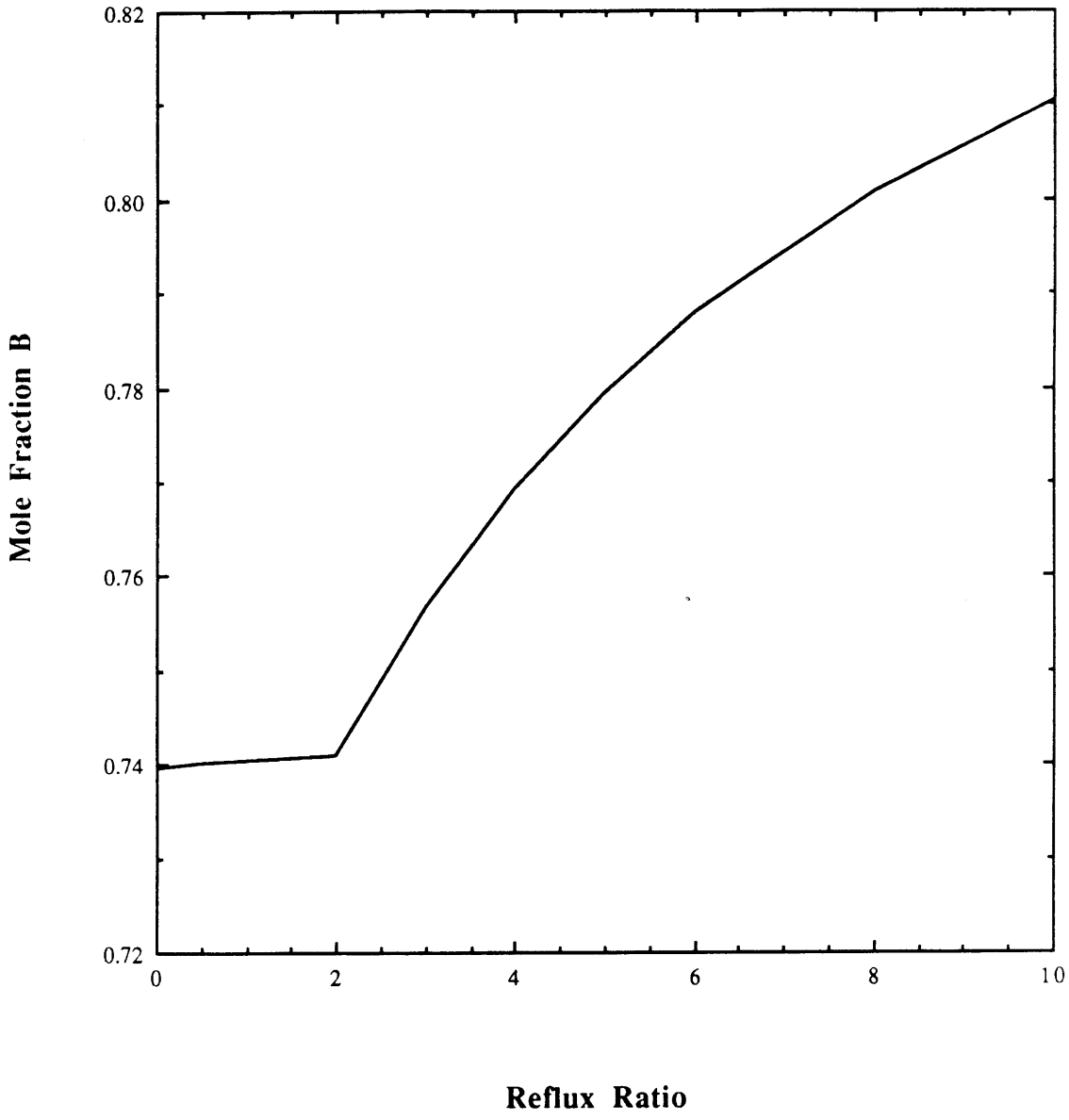


FIGURE 4-18. PERFORMANCE LOAD SHIFT: OPTIMAL x_B AS A FUNCTION OF REFLUX RATIO

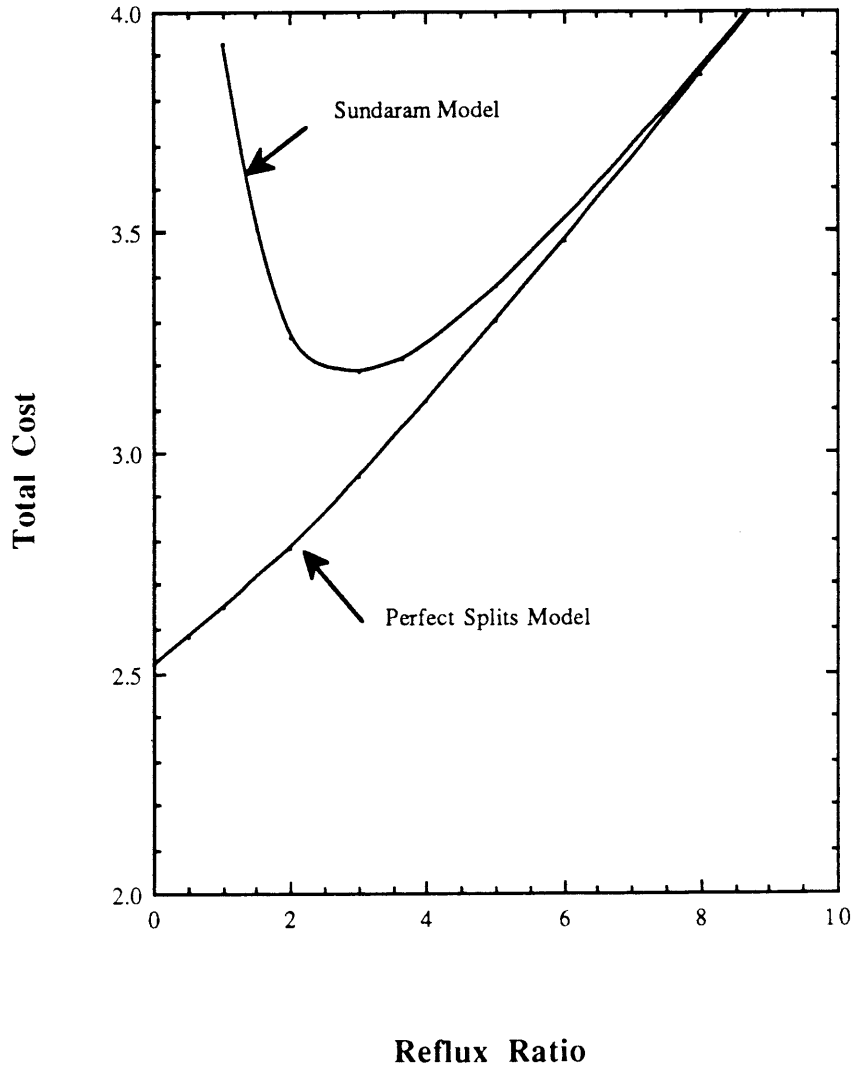


FIGURE 4-19. COMPARISON OF PERFECT SPLITS AND SUNDARAM COLUMN MODELS: OPTIMAL COST AS A FUNCTION OF REFLUX RATIO

Example Problem #3 has shown the effects of model complexity on the performance of the solution procedure. Large increases in computation time (one to three orders of magnitude higher) have been observed for more-detailed process models, particularly when differential equations are integrated numerically. Although optimal solutions are obtained for these somewhat more complicated example problems, the rapid increase in computational effort does not bode well for applying this mathematical programming solution approach to larger problems.

4.6 Multiple Products

All of the example problems described in Section 4.5 dealt with processes that produced only one product. The NLP formulation given for the performance subproblem in Section 4.3 includes the possibility of producing more than one product on the given equipment units. This section briefly describes some of the effects that multiple products have on the optimization problem.

The main change in the problem is the increased number of decision variables. Since the performance of each process has to be optimized, the number of decision variables increases roughly linearly with the number of products. Multiple products have adverse effects on the solution of the performance subproblem for two reasons. First, the larger number of decision variables slows down the convergence of the SQP algorithm. Second, the solution time for evaluating the objective function increases because the process must be simulated for each product on each iteration.

Applying the SQP algorithm to the NLP formulation for multiple products causes the process operating conditions for each product to be optimized simultaneously. An alternative approach is proposed in an attempt to reduce the negative effect on the computation time caused by increasing the number of products. Since the campaign change-over times have been neglected, the only way that the various products interact is through the overall time horizon constraint. By assigning some portion of the horizon time to each product, the products can be optimized individually. These sub-horizon times for each product would have to be adjusted to satisfy the overall horizon constraint and minimize the overall cost.

An approximate procedure for getting near-optimal solutions is described below. The general approach consists of three steps, as shown in Figure 4-20. First, upper and lower bounds on the campaign time for each product are determined. Second, an

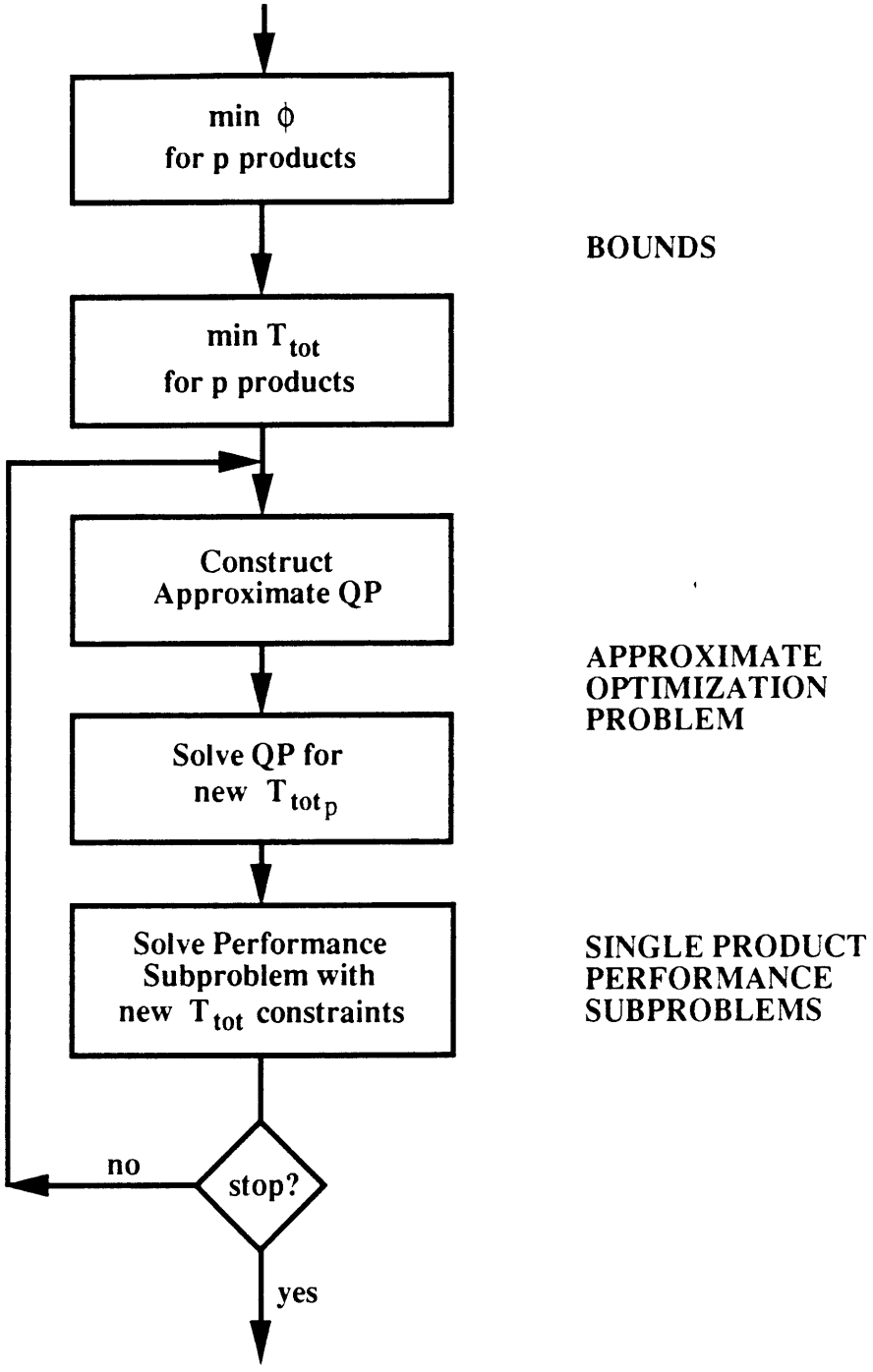


FIGURE 4-20. APPROXIMATE METHOD FOR MULTIPLE PRODUCTS

approximate quadratic programming problem is solved to estimate the sub-horizon times for the individual products. Then, single product performance subproblems are solved to determine the actual operating conditions and costs for each product for the specified sub-horizon times. The second and third steps can be repeated in an iterative fashion until some user specified stop criterion is satisfied. This criterion might be percentage cost reduction from iteration to iteration or proximity to a bound on the minimum cost.

Bounds on the campaign time for each product are determined first. The upper bound corresponds to the campaign time when the total operating costs (including rental charges) are minimized. The lower bound is the minimum campaign time in which the specified amount of product can be made. This second case will have higher operating costs. To obtain these bounds for all the products will require the solution of 2P single product performance subproblems.

These bounds can be used to reduce the amount of effort spent on the multiproduct optimization problem. If the sum of the upper bound campaign times for all the products is less than the overall horizon time, then the original multiproduct optimization problem is solved. Each product should be produced with its lowest cost operating conditions. If the sum of the lower bounds (minimum campaign times) exceeds the overall horizon time, then there is no feasible solution to the overall problem with the given equipment units. In either of these cases, there is no point in doing any additional work.

Assuming that the problem has not yet been solved, the bounds on the campaign times are then used to develop estimates for the sub-horizon times for each product. A typical plot of optimal cost as a function of campaign time for a single product is shown in Figure 4-21. Determining the exact points in this curve would require the solution of a single product performance subproblem with a horizon constraint at each possible value of the campaign time. This excessive amount of work can be avoided by constructing a quadratic approximation to the curve. For the first iteration, only the two end points are known (from the bounds). By assuming that the slope of the fitted curve is zero at the upper bound for the campaign time, a quadratic cost function in terms of the campaign time can still be constructed for each product. On subsequent iterations, a best fit quadratic function is fitted using all actual data on the cost as a function of the campaign time for each product.

A quadratic programming (QP) problem is formulated to determine the "best" values of the campaign times. The QP objective function consists of the sum of the

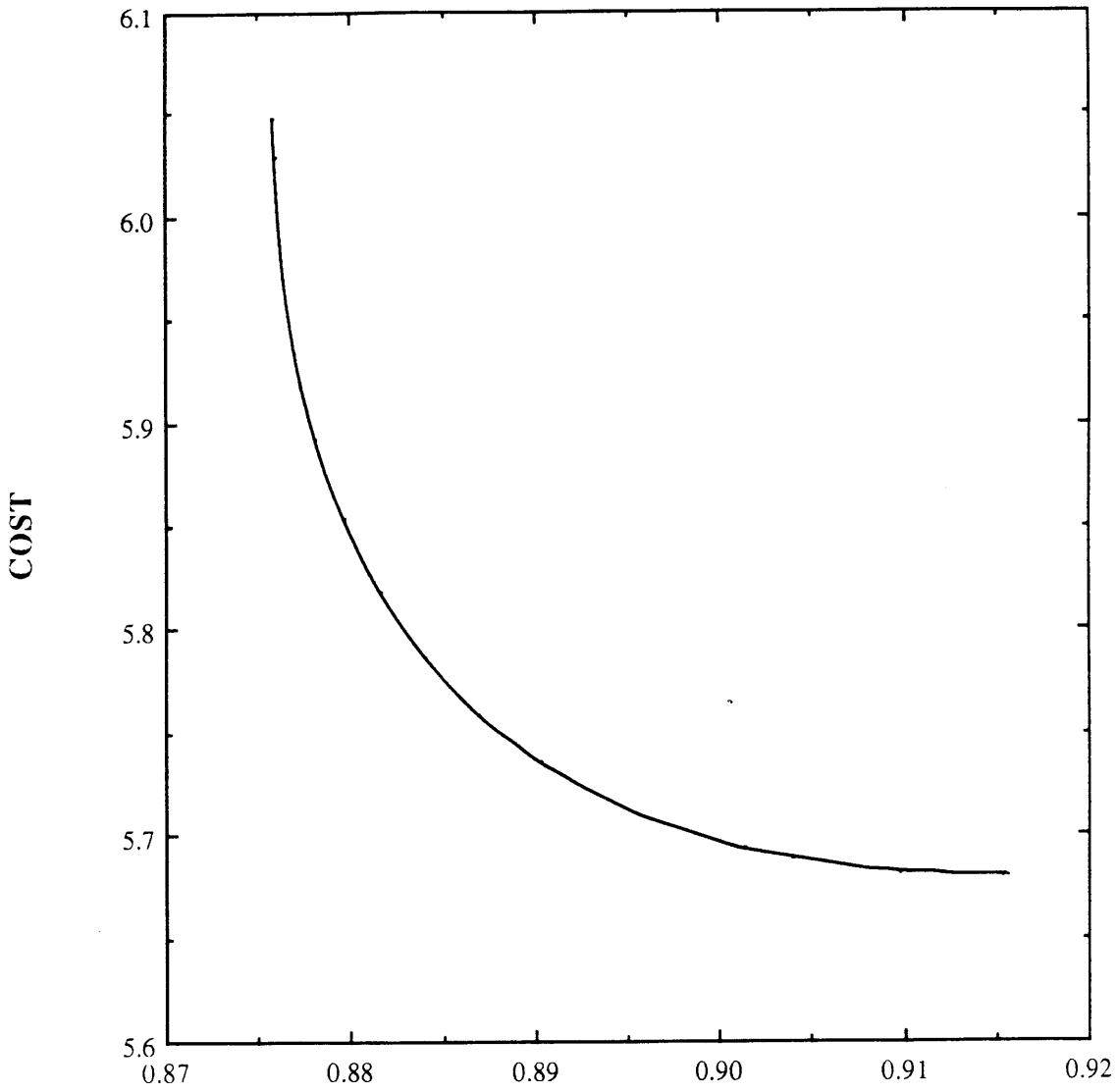


FIGURE 4-21. TYPICAL PLOT OF OPTIMAL OPERATING COST AS A FUNCTION OF CAMPAIGN TIME

approximated quadratic cost functions. The horizon constraint is a linear inequality constraint in terms of the product campaign times. The bounds obtained in the first step are imposed as constraints on the P campaign times. This simple QP requires very little computational effort to solve.

The campaign times obtained from the QP are used as the new sub-horizon times. The single product performance subproblems are re-solved with the new sub-horizon times to obtain the actual operating conditions and costs. Summing over the products, the actual total cost is obtained, and the stopping conditions are checked. If these conditions are not satisfied, the second step is repeated by fitting new quadratic cost approximations. Steps two and three are then repeated until the stopping criteria are satisfied.

The main benefit of using the approximate strategy described above rather than simultaneously solving for the optimal conditions using the multiproduct NLP formulation for the performance subproblem is that only simpler, single product optimization problems need be solved. Single product performance subproblems have fewer optimization variables and would be expected to converge faster than the larger multiproduct formulations. Also, by using the approximate quadratic representation, fewer simulations are required. The key trade-off is between solution accuracy and computation time.

4.7 Implementation Issues

4.7.1 Overview

There are a number of potential difficulties in implementing a mathematical programming solution approach to larger problems of industrial interest. The three problems that appear to cast the largest shadows over this approach at the present time are (1) the availability of adequate models and process data, (2) the potentially large computation time caused in part by the differential-algebraic nature of the process models, and (3) possible numerical problems caused by feasible regions that are in general non-convex. Some thoughts on how these problems might affect the implementation of this solution approach on larger problems are given in this section. In addition, some ideas are also presented on possible ways to minimize the effects of these potential problems.

4.7.2 Availability of Process Models

The first potential problem mentioned above is the availability of models and process data. Particularly at the design stage, it is unlikely that much information is

available for a new specialty product. Part of the higher value associated with these products often comes in the form of new and unique properties. Detailed kinetic mechanisms with known values for rate constants may not yet be available. Just being able to consistently make the product so that it can be introduced to the market before some competitor might be all that management really wants. The development of a good process model and the data required to get it to match reality may be seen as a waste of time and money considering the relatively short expected lifetime of the product. Also, since the product is valuable because of its unique properties, no one wants to take the chance of messing up these properties by altering the chemist's original recipe. Therefore, the need for good models is often not perceived, and consequently, models are not developed.

However, good models can definitely provide benefits in the form of more efficient plant operation and higher profit margins. As the benefits of using computer methods to improve process operation and speed up the design process are demonstrated, the number of models will increase. Parakrama (1985) noted that industry desired computer capabilities to optimize process operating conditions. To accomplish this task, better and more abundant process models will be required. These models might be based on first principles, or they could simply be empirical relationships obtained from pilot plant experiments. The appearance of a commercial simulator (BATCHES) is further evidence of the trend towards better modeling of batch processes.

As more models become available, better efforts will be made to obtain the processing data necessary to use them effectively. Group contribution methods are already being used by continuous process simulators to estimate the physical properties of new materials. These methods could also be used for specialty materials that are to be produced in batch processes.

4.7.3 Computational Load

The second problem mentioned above concerns the potentially large computation time required by the optimization problem. The large number of decision variables in the optimization and the large number of calculations done by the process model are the two biggest potential causes of excessive computational load. This problem could be especially troublesome for large complex processes, particularly when the number of products to be produced gets large, or when the process model requires the integration of stiff systems of differential equations.

The number of decisions is approximately the number of degrees of freedom since equality constraints are being handled internally by the process models. A larger number of decision variables means that a larger number of iterations are required by the optimization routine to converge. In addition, variable scaling problems could become more difficult to diagnose and fix as the number of decisions increases. Also, each extra optimization variable generates one additional simulation on each iteration in order to get the necessary derivative.

One possible way to address this problem is to reduce the total number of decision variables down to some smaller number of key decision variables. This approach was used by Fisher et al. (1985) to generate roughly "optimized" alternatives in a synthesis methodology for continuous plants. By doing a sensitivity analysis, a small number of key variables were selected for optimization. Non-key decision variables were fixed at nominal values. The resulting smaller optimization problem was quickly solved to give a better estimate of the actual optimum. The same principle could be applied to batch processes to reduce the number of decision variables. The cycle of sensitivity analysis and optimization could be repeated until diminishing returns are obtained.

Another way to reduce the computational effort would be the use of simpler, short-cut models in the optimization. Simultaneous modular approaches (Boston and Britt, 1978; Jirapongphan et al., 1980; Jirapongphan, 1980; Evans et al, 1985; Trevino, 1985; Ganesh and Biegler, 1987) have used this idea to reduce the computation times for optimizing continuous process flowsheets. Short-cut models might remove some of the complexity from detailed models in order to produce models that predict the most important aspects of the process behavior at a fraction of the computational effort. These reduced models might be used in conjunction with the more rigorous models in some type of iterative fashion. The short-cut models might be generated by doing a small number of simulations around a base point and fitting the required model constants. This simpler model would then be used in the optimization. Substantial savings in computation time might be possible if analytic derivatives can be derived for the short-cut models or if the need to integrate differential equations can be eliminated. Further research needs to be done in this area to develop optimization schemes that make use of reduced models in the performance subproblem.

Collocation schemes are an example of the short-cut models just discussed. Differential-algebraic systems have been studied by a number of workers (Cuthrell and

Biegler, 1985, 1986; Renfro et al., 1987). These investigators have addressed the issue of reducing computational effort for optimizing continuous processes by using collocation methods to eliminate the need to integrate differential equations. The flowsheet equations and polynomial approximations for the differential equations are converged simultaneously with the optimization. Thus, the differential equations are replaced with algebraic equations in the NLP. The number of collocation points is an important issue in these methods because this number affects both the accuracy of the approximation and the number of variables in the new NLP.

A final way to overcome large computation times is the use of faster, more powerful computers. The increased availability of extremely powerful computers has made possible the solution of many problems which would have been deemed beyond the capabilities of even the largest computers just ten years ago. The advent of low cost, high performance computing capabilities will certainly make the solution of the performance subproblem more tractable for the design engineer.

4.7.4 Non-Convexities

The third problem mentioned above concerned potential numerical problems caused by non-convexities in the feasible region. This problem is a general one that must be considered whenever mathematical programming methods are employed. A non-convex feasible region means that there is no guarantee that a global optimum solution will be obtained by the optimization method. Non-convexities are very likely to show up in practical industrial problems because of the complexity of process models and nonlinear inequality constraints.

One strategy for obtaining a global optimum is the brute force method. A series of different starting points is used, and the best solution found is assumed to be close to the global optimum. Unfortunately, it is not usually easy to determine how close this best solution is to the global optimum. Also, there is no useful guide for determining how many starting points should be tried. Recently, a number of researchers (Lucia, 1987; Kocis and Grossmann, 1988; Aggarwal et al., 1988) have begun addressing the problem of finding globally optimal solutions to non-convex optimization problems. Future work will need to try to apply some of these more promising methods to non-convex performance subproblems.

4.8 Chapter Summary

This chapter has focused on the performance subproblem. A number of generic performance trade-offs have been identified and discussed. The optimization problem has been formulated as an NLP, and a number of example problems have been solved using a SQP solution strategy. These problems clearly illustrated the benefits of exploring performance trade-offs to reduce process operating costs. The effect of the overall plant environment on the optimal processing conditions (through the setting of the rental charges) has also been demonstrated. Short-cut procedures have been developed for handling the multiproduct design case. Finally, high computational loads have been identified as a major potential problem for applying a mathematical programming solution procedure to larger problems.

Chapter 5

STRUCTURE SUBPROBLEM

5.1 Introduction

The structure subproblem deals only with task (c) of the design tasks described in Section 2.2.1. Decisions are assumed to have been made already for the other four tasks, and thus the overall operating mode, the task to stage assignments, the locations of intermediate storage, and the process operating variables and times have been fixed. The only trade-offs left to be investigated concern the choices of unit sizes and the number of units in parallel at each stage.

The optimal sizing problem has been studied extensively by previous workers as described in Section 2.2.2. However, only a few workers (Sparrow et al., 1975; Papageorgaki and Reklaitis, 1989) in the batch design area have assumed that units are only available in discrete sizes. The major difference between these previous efforts and this work is that the focus here is on the design of a process that will be incorporated into an existing plant. The objective function of minimizing operating costs contrasts sharply with the minimization of capital cost for a new grass roots plant that has been used in previous work.

Since the new process is to be carried out in an existing facility, equipment units have distinct sizes. Also, only a finite number of units are available. Parallel units may not always be identical. The cost of using the equipment units is incorporated into the objective function through the use of equipment usage (or rental) charges. For given process performance, this version of the optimal sizing problem involves only discrete decisions. Thus, a combinatorial optimization problem is encountered.

This chapter describes the structure subproblem. First, the generic types of structure trade-offs that must be considered are reviewed. Next, the overall problem formulation from Chapter 3 is simplified to handle this subproblem. Although "exact" solution approaches are available, an approximate solution strategy is developed in order to reduce computational requirements. Example problems are solved to demonstrate the

various characteristics of the problem and show the application of the approximate solution approaches. Finally, some comments are made regarding the extension of these methods to multiple products.

5.2 Structure Trade-offs

Structural trade-offs have been considered by the many previous workers on the optimal sizing problem and its extensions. These trade-offs included the relative sizing of processing units, the number of units in parallel at each stage, the task to stage assignments, and the location of intermediate storage. These aspects of batch processes were reviewed in Section 2.1.2. Because these various structural decisions affect the sizes of the process units, and hence the total capital cost, structural trade-offs were explored to determine the minimum capital cost plant.

For the structure subproblem to be described here, only "sizing" issues and the number of units in parallel are considered. The task to stage assignments and the locations of intermediate storage are assumed to be given. Because existing equipment units are to be used, the relative sizing of units manifests itself in the assignment of discrete units to processing stages. The number of parallel units at a stage is also determined by the assignment of units to stages. Generic structural trade-offs are reviewed here since these trade-offs are the focus of the structure subproblem. These structural trade-offs appear in the form of discrete alternatives.

Each possible assignment of units to stages has its particular bottleneck stage which determines the average production rate for the process. Since the process performance is fixed for the structure subproblem, the only cost term in the objective function that can vary is the cost for using the equipment. The total equipment cost depends on the rental charge rates for the units selected from the inventory of available and the total time that the units are tied up, i.e., the campaign time. The campaign time is determined by the bottleneck stage average production rate.

There are three steps that must be carried out when solving the structure subproblem. The choices made during these steps determines the quality of the final solution. First, the number of units to be selected from the inventory of available units must be determined. Second, this number of individual units must be chosen. Finally, each of the units picked must be assigned to a particular processing stage. In each of these three steps, many discrete alternatives are possible.

A simple example is given to illustrate these alternatives. Consider a two stage batch process consisting of reactant preparation followed by reaction. Both stages require agitated, jacketed vessels. Although both stages have the same time requirement of four hours, the second stage has a larger size factor because the second reactant is added directly to the reactor. Four tanks are available in the inventory. The sizes are given in Table 5-1 along with the average rate for each unit at the two possible stage locations.

The first step involves selecting the number of units to be used. Since each stage must have at least one unit, the choices are two, three, and four. In general, the number of alternatives is given by the difference between the number of units available and the number of stages plus one.

The second step involves choosing the specific units to be utilized. The number of alternatives depends on the number of units to be selected and the number available. The number of distinct sets of units, n, (assuming no identical units) is given by:

$$n = C_m^k \tag{5-1}$$

where k is the number of available units, and m is the number of units chosen. If identical units are available, this formula will over-predict the number of sets. Suppose three units are to be selected in the two stage example. Then, there are four possible sets of three units that could be selected: {1,2,3}, {1,2,4}, {1,3,4}, and {2,3,4}.

The third step involves assigning the units to the processing stages. Once the particular units have been selected, the total rental rate for the process is known. In order to minimize the equipment costs, the units should be assigned to maximize the bottleneck rate (and thus minimize the campaign time). Again, a number of discrete alternatives exist. Suppose set {2,3,4} is selected at step two. Two units could be assigned to the preparation stage and only one to the reactor stage. Three distinct structures are possible with this arrangement, depending which unit is assigned to the second stage. Alternatively, only one unit could be placed at the first stage. This approach generates three additional possible structures.

The specific unit to stage assignments determine the bottleneck rate, the total equipment rental rate, and the total equipment cost. It is beneficial to assign parallel units to the bottleneck stage in order to increase the overall average production rate. Assigning additional units to non-bottleneck stages increases the total equipment cost without any

TABLE 5-1.
STRUCTURE TRADE-OFFS: DATA FOR SIMPLE EXAMPLE

Vessel	Size (l)	Rental Rate ($ / hr)	Average Rate (kg / hr) Stage 1	Stage 2
1	500	20	50	30
2	750	28	75	45
3	1250	37	125	75
4	2000	55	200	120

benefit of increased production rate. The average rates and total cost for the six possible structures for set {2,3,4} are summarized in Table 5-2.

5.3 MINLP Formulation

The general mathematical problem formulation developed in Section 3.3.3 for minimizing the total operating costs of a multiproduct batch plant can be greatly simplified when the process performance is fixed. However, the problem remains a mixed integer nonlinear programming problem (MINLP). The MINLP for the structure subproblem has the binary variables (y_{ijk}), which represent the locations of the available units, as the decision variables in the optimization.

The overall optimization problem essentially reduces to an assignment problem, where available units are either assigned to a stage in the process or left in inventory. Figure 5-1 shows a graphical representation of all possible assignments of units to stages. For any given structure, only some of all the possible arcs will exist. Binary variables y_{ijk} represent the presence of unit k of type j at stage i of type j. Constraints on this assignment procedure include the requirements that every stage have at least one unit and that no unit may be assigned to more than one stage.

The MINLP for the structure subproblem consists of minimizing the objective function given by Equation 3-2 in Section 3.3.3 subject to a number of constraints. The objective function includes all operating costs for producing specified product amounts, although fixed process performance means that the raw materials, waste treatment, and utilities costs are essentially constant. Only the equipment usage costs are affected significantly by varying the structure.

The complete set of constraints includes the constraints on the binary variables given by Equations 3-6, 3-9, and 3-10, and the constraints for determining cycle times, batch sizes, and average production rates given by Equations 3-11 to 3-17. Recall that this set of constraints for determining the average production rates assumes that intermediate storage is available between all stages. Variable bounds are given by Equations 3-18 through 3-21. Because the operation times ($\underline{t}$) and the operating variables ($\underline{x}$) are fixed, it is assumed that the process model equations are implicitly satisfied, thus eliminating the need to include Equations 3-3, 3-4, and 3-5 in the constraint set.

If all the units are assumed to run full, the MINLP can be rewritten in an even more reduced form. First, Equation 3-12 can be recast as an equality constraint and solved for

TABLE 5-2.
STRUCTURE TRADE-OFFS: RESULTS FOR 6 POSSIBLE STRUCTURES USING VESSELS 2, 3, AND 4

STRUCTURE		RESULTS*		
Stage 1	Stage 2	Bottleneck Average Rate (kg / hr)	Campaign Time (hr)	Total Equipment Cost ($K)
2	3,4	75	667	80
3	2,4	125	400	48
4	2,3	120	417	50
3,4	2	45	1111	133
2,4	3	75	667	80
2,3	4	120	417	50

* Campaign produces 50,000 kg of product (UIS policy).

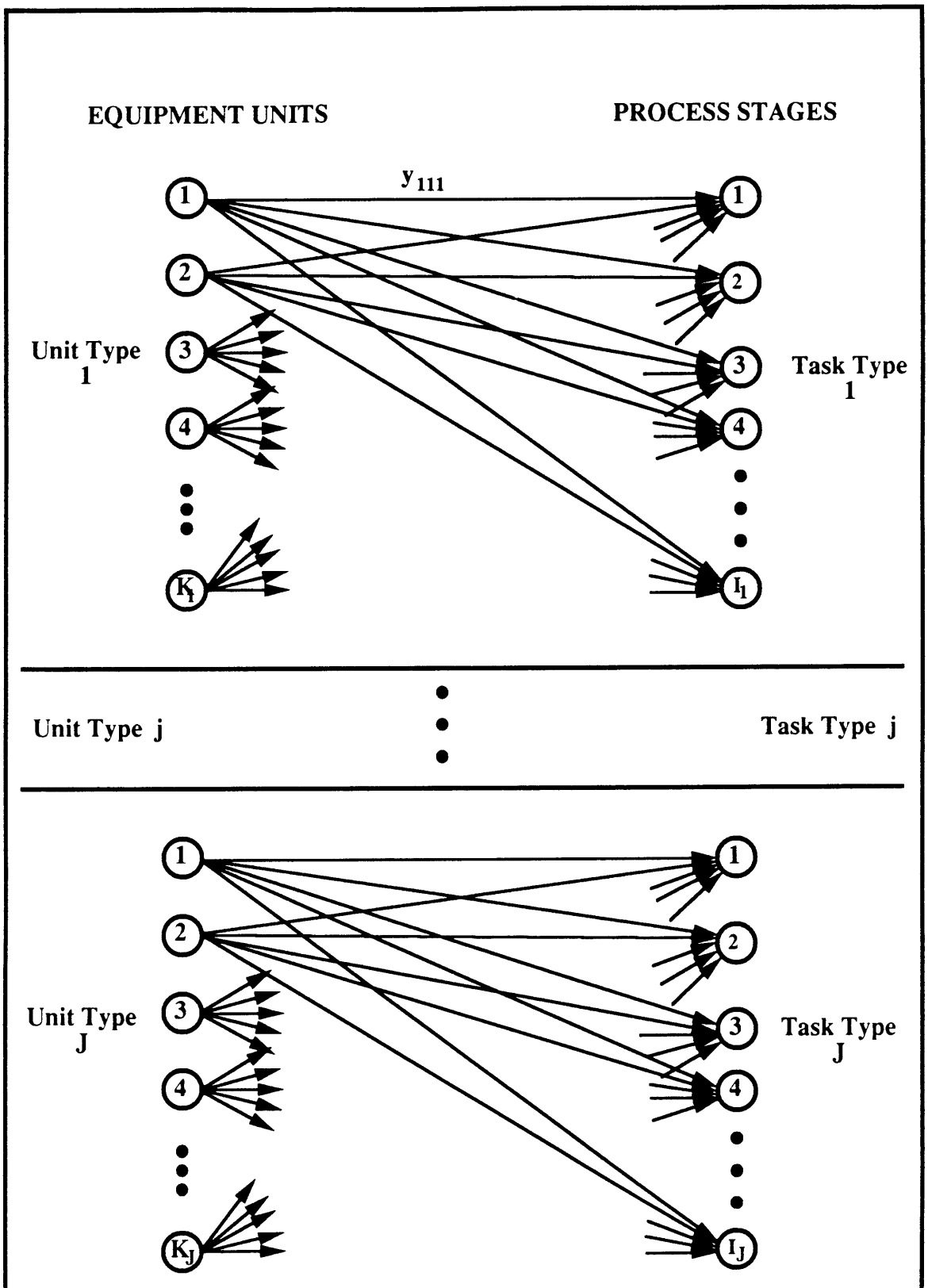


FIGURE 5-1. EQUIPMENT ASSIGNMENT PROCESS

the batch size of unit k at all possible stages ij for each product p:

$$B_{ijkp} = \frac{V_{jk}}{S_{ijp}} \qquad \text{for all i,j,k; p=1,P} \qquad (5\text{-}2)$$

Knowing the batch sizes B_{ijkp} means that the cycle times T_{ijkp} can be calculated directly from Equation 3-11. The batch sizes and cycle times are then used in Equation 3-13 to calculate the average rate r_{ijkp} for all possible unit to stage assignments. These calculations can be done in advance when the process performance is fixed.

The reduced MINLP formulation for the structure subproblem is shown below:

$$\min \phi = \sum_{p=1}^{P} (\phi_{op} + E \frac{A_p}{R_{L_p}}) \qquad (5\text{-}3)$$

$$E - \sum_{j=1}^{J} \sum_{k=1}^{K_j} \sum_{i=1}^{I_j} y_{ijk}\, e_{jk} \geq 0 \qquad (5\text{-}4)$$

$$\sum_{k=1}^{K_j} r_{ijkp}\, y_{ijk} - R_{L_p} \geq 0 \qquad \text{for all i,j; p=1,P} \qquad (5\text{-}5)$$

$$\sum_{i=1}^{I_j} y_{ijk} \leq 1 \qquad \text{for all j,k} \qquad (3\text{-}9)$$

$$\sum_{k=1}^{K_j} y_{ijk} \geq 1 \qquad \text{for all i,j} \qquad (3\text{-}10)$$

$$\sum_{p=1}^{P} \frac{A_p}{R_{L_p}} \leq H \qquad (5\text{-}6)$$

Given parameters include the equipment usage charge rates for each unit, e_{jk}, the amount of each product that must be produced, A_p, and the overall time horizon, H. The bottleneck stage processing rates R_{Lp} and the total equipment rental rate E are the only continuous variables in the reduced formulation. The objective function, ϕ, consists of two terms as shown in Equation 5-3. The first term, ϕ_{op}, is a constant that includes operating cost items that are fixed when operating variables and processing times are specified. The second term contains the total usage cost for tying up the equipment units during the length of the

campaign. Equation 5-6 for the horizon constraint is simply a combination of Equations 3-16 and 3-17.

Since the total cost is inversely proportional to the bottleneck stage processing rates, increasing the bottleneck rates will reduce the total cost for a process if the total equipment rental rate E does not increase too much. For a specified set of units, the numerator in Equation 5-3 is fixed. Then, by assigning the units to the "proper"stages to get high bottleneck rates, a lower total cost is obtained. Another possible way to increase the bottleneck rates would be to add parallel units to the slow stages. However, additional units increase the total equipment usage charge rate. Thus, the addition of parallel units increases both the numerator and the denominator in Equation 5-3, and the resulting effect on the overall cost cannot be predicted *a priori*. Because of the discrete nature of the problem, the cost is expected to be a discontinuous, non-unimodal function of the bottleneck rate. The quotient in the objective function also causes the problem to be nonlinear.

The total number of binary variables in this reduced MINLP formulation is given by:

$$N_{bvar} = \sum_{j=1}^{J} I_j K_j \tag{3-7}$$

The total number of continuous variables is given by:

$$N_{cvar} = P + 1 \tag{5-7}$$

The total number of constraints is given by the following expression:

$$N_{con} = (\sum_{j=1}^{J} (P + 1)I_j + K_j) + 2 \tag{5-8}$$

A single product process with two types of tasks (J=2), two processing stages of each type ($\underline{I}$=(2,2)), and four and six units available ($\underline{K}$=(4,6)) generates an MINLP with 20 binary decision variables, 2 continuous variables, and 20 linear constraints. If three products are produced rather than just one, the MINLP would still have 20 binary decisions. However, the number of continuous variables would increase to 4, and the number of linear constraints would be 28. Thus, the number of discrete alternatives is independent of the number of products to be produced on the equipment.

The total number of combinations of y_{ijk}'s would be over 1 million for the example just cited. Because the total number of combinations doubles with each additional binary decision variable, the combinatorial problem becomes very difficult to solve exactly for the global optimum for even moderately sized problems. The number of combinations of binary variables exceeds the number of actual structures because some combinations violate constraints and thus are not feasible structures. For example, one possible combination would have all the y_{ijk}'s equal to zero, which is clearly not feasible. However, the number of actual structures can still be quite large. As shown in the next section, calculating this number is quite involved.

5.4 Determining the Number of Possible Structures

The difficulties in determining the total number of possible structures arise from the large number of alternatives at each of the three levels of decisions that must be made when making a structure assignment. As discussed in Section 5.2, these three steps are (1) the choice of the number of units, M, to be used (or assigned), (2) the selection of a set of M units, and (3) the assignment of the M selected units to the process stages. Two simple examples are given in order to show how these aspects affect the determination of the number of possible structures.

First, assume that three units (different sizes) are available for a one stage process. Allowing any number of units to be used in parallel, there are a total of seven possible structures as shown in Figure 5-2. If only one unit is assigned, there are three choices. For two units in parallel, there are also three ways to select pairs of units. For three units in parallel, there is obviously only one structure possible. Summing up these three different cases gives a total of seven structures.

For the second example, assume that there are now two process stages to fill with the inventory of three units. See Figure 5-3. In the first example, one, two, or three units could be used in the process. In this example, only two or three units can be used because each stage must have at least one unit. If two units are used, there is one unit at each stage. There are again three ways to pick the pair of units to be used in the process, but now the order matters because there are two stages to fill. Thus, for this case there are six structures. If three units are used, there will be one stage with two units in parallel and one stage with only one unit. Assuming that the parallel units are at stage 1, there are three ways to select the one unit to put at stage 2. Of course, the location of the two parallel units

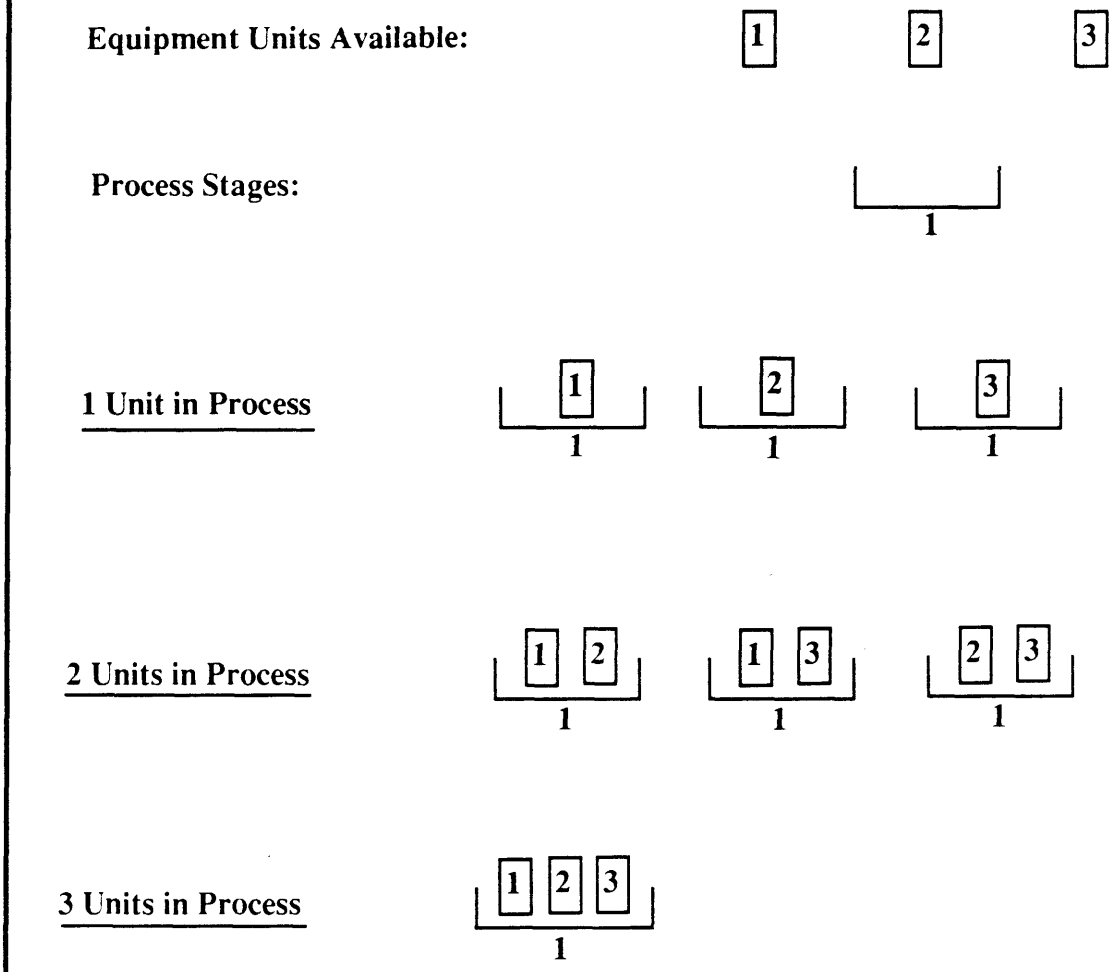


FIGURE 5-2. NUMBER OF STRUCTURES: 1 STAGE PROCESS

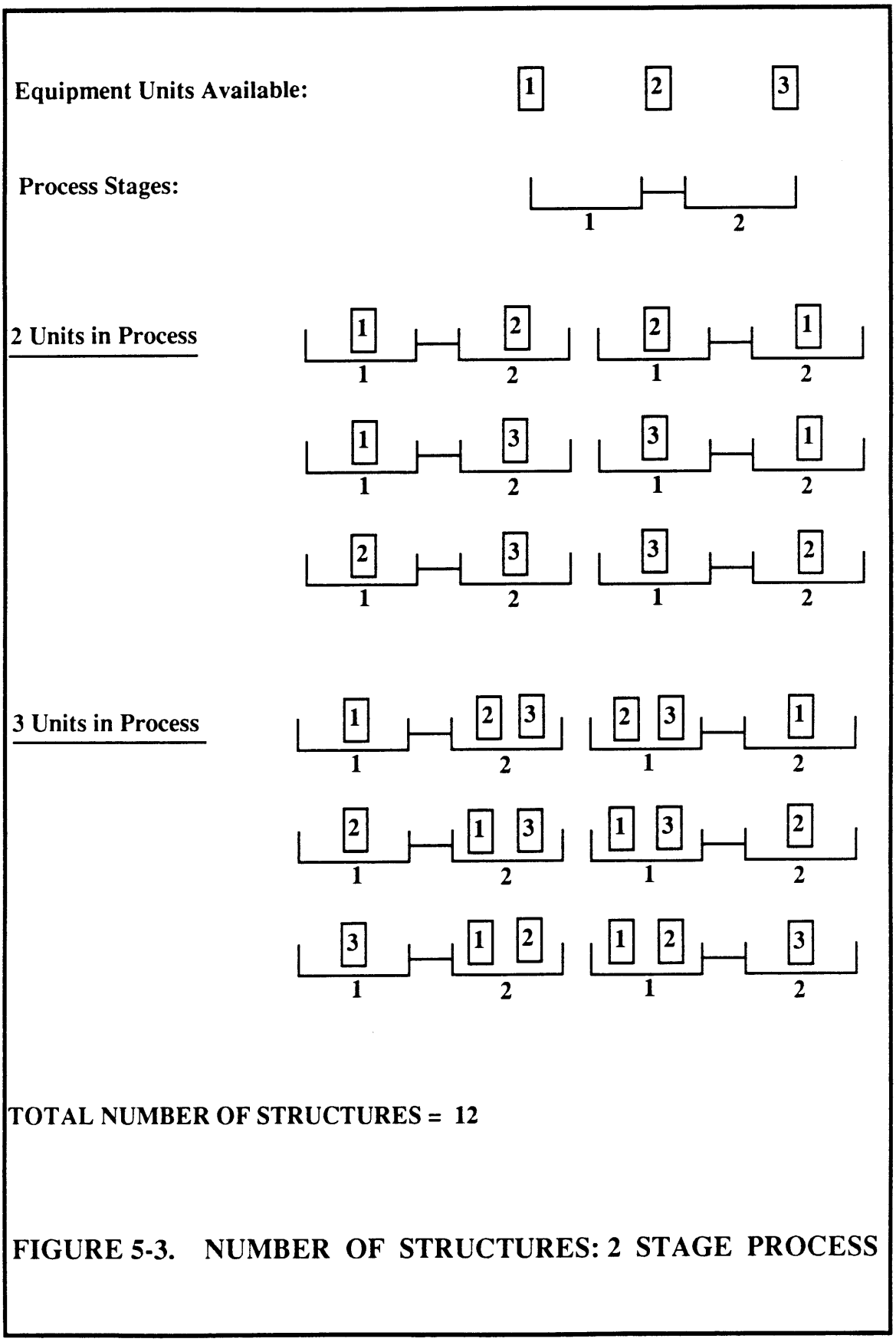


FIGURE 5-3. NUMBER OF STRUCTURES: 2 STAGE PROCESS

could be at stage 2, which would give three additional structures. Thus, for the second example, there are a total of twelve structures.

A general method for determining the total number of possible structures is described below. One key assumption is that there are no identical units. If identical units are available, some structures will be double counted. It is assumed that the process consists of I_j stages for each of J types of tasks. The equipment inventory has K_j units of J types available for use. A one-to-one matching between equipment and task types is also assumed.

The procedure for determining the total number of structures involves a series of nested calculations. The general flow of the method is given in Figure 5-4. At the lowest level, the number of structures possible for a given arrangement must be calculated. Next, the total number of possible structures for a specified number of units in the process is determined by summing up the number of structures for all possible arrangements. Since the number of units assigned from the inventory to the process can vary from the number of stages to the total number of units available, the total number of structures must be calculated for all possible numbers of units in the process for each type of processing stage. Finally, the grand total is obtained by taking the product of the number of structures for each type. Details are provided for each step in the procedure in the following paragraphs.

An ***arrangement*** is a particular way that m elements have been divided into i groups. In this case, m is the number of units and i is the number of stages. It is not the particular element to group assignments that define an arrangement, but rather the number of elements assigned to each group. Consider the case where the elements {A,B,C,D,E} must be partitioned in three groups. In this case, m equals five, and i equals three. An ***assignment*** denoted by (A/B/C,D,E) indicates that element A has been assigned to the first group, element B has been assigned to the second group, and that elements C, D, and E have been assigned to the third group. The assignments (A,B/C,D/E) and (D,E/A,B/C) are both structures in the arrangement [2/2/1], because two elements have been assigned to the first group, two elements have been assigned to the second group, and one element has been placed in the third group.

The number of distinct assignments of m elements to i groups for a given arrangement $[u_1/u_2/\ldots/u_i]$ shall be denoted by η_{mia}, where the subscript "a" indicates arrangement number "a". The vector $\underline{u}$ contains the number of elements in each group for a

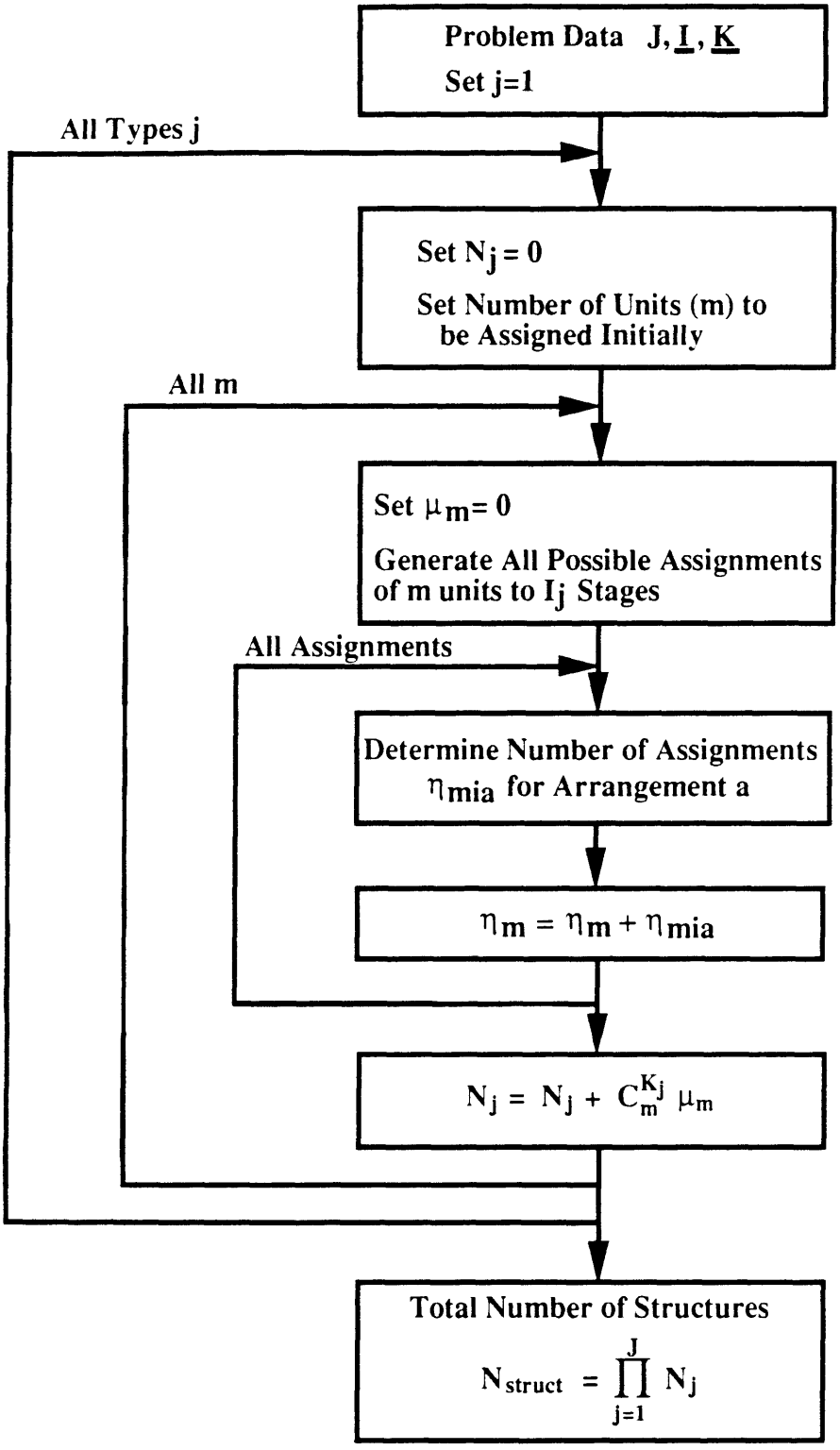


FIGURE 5-4. LOGIC DIAGRAM FOR CALCULATING THE NUMBER OF POSSIBLE STRUCTURES FOR GIVEN VALUES OF J, I, AND K

given arrangement. The value of η_{mia} is given by:

$$\eta_{mia} = \prod_{k=1}^{i} C_{U_k}^{m+u_k-\sum_{n=1}^{k} u_n} \tag{5-9}$$

For the [2/2/1] arrangement shown above, a total of 30 unique assignments exist.

The number of possible arrangements, A_{mi}, of m elements in i groups is given by:

$$A_{mi} = C_{i-1}^{m-1} \tag{5-10}$$

For the example above with five elements and three groups, a total of six arrangements exist. Determining the complete set of arrangements (and thus the $\underline{u}$ vectors) is considerably more difficult than calculating the number of assignments.

The $\underline{u}$ vectors for all possible arrangements are obtained using the following scheme. A set of nested loops is constructed. The upper limit, n_k, of the kth loop counter is given by:

$$n_k = m - i + k + u_k - \sum_{b=1}^{k} u_b \qquad \text{for } k=1, i-1 \tag{5-11}$$

On each iteration, the u_k's are set equal to the current values of the corresponding loop counters, with the exception of u_i. The number of elements in the last group, u_i, is determined by:

$$u_i = m - \sum_{k=1}^{i-1} u_k \tag{5-12}$$

An arrangement counter a is incremented every time Equation 5-12 is executed. Thus, each individual set of $\underline{u}$ values is associated with an arrangement number. Figure 5-5 shows the calculational procedure for determining the $\underline{u}$ vectors for the case when m=5 and i=3. Table 5-3 shows the six possible arrangements and the total number of assignments for each for the simple example described above.

The total number of assignments of m elements to i groups for all possible arrangements, μ_m, is obtained by summing over all the arrangements:

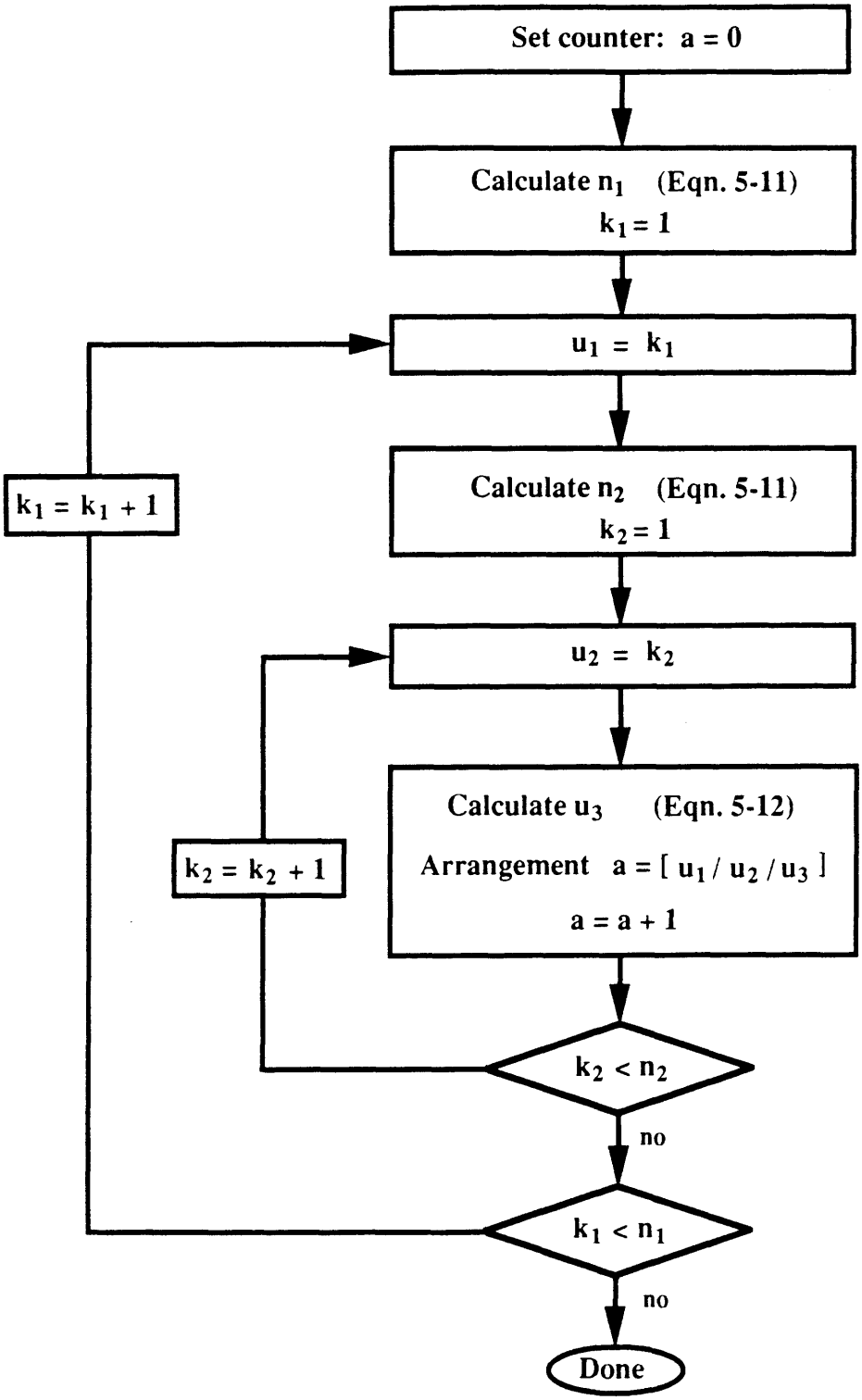


FIGURE 5-5. EXAMPLE OF CALCULATIONAL PROCEDURE FOR DETERMINING ALL POSSIBLE ARRANGEMENTS (m=5, i=3)

TABLE 5-3.
LIST OF 6 POSSIBLE ARRANGEMENTS FOR 5 UNITS BEING ASSIGNED TO 3 STAGES

Arrangement [u_1 / u_2 / u_3]	Number of Possible Assignments for Given Arrangement
[1 / 1 / 3]	20
[1 / 2 / 2]	30
[1 / 3 / 1]	20
[2 / 1 / 2]	30
[2 / 2 / 1]	30
[3 / 1 / 1]	20
Total Number of Assignments for the 6 Arrangements:	150

$$\mu_m = \sum_{a=1}^{A_{mi}} \eta_{mia} \tag{5-13}$$

In general, the number of units available in the equipment inventory (K_j) for unit type j is greater than the number of stages (I_j) of task type j. Thus, the number of units of type j actually used in the process, m_j, could vary from I_j to K_j. The fact that a number of different sets of m_j units can be selected from the K_j available must also be taken into account. For K_j units and I_j stages, the total number of structures, N_j, is given by:

$$N_j = \sum_{m=I_j}^{K_j} C_m^{K_j} \mu_m \tag{5-14}$$

Since each unit type can only be assigned to one task type, the grand total number of structures for a process of J types of tasks, I_j stages, and K_j units is given by:

$$N_{struct} = \prod_{j=1}^{J} N_j \tag{5-15}$$

Table 5-4 shows the number of structures N_j for a number of combinations of I_j and K_j. Note that the number of structures grows explosively with problem size. A single product process with two types of tasks (J=2) with two stages of each type ($\underline{I}$=(2,2)) and four and six units available ($\underline{K}$=(4,6)) would have 30,100 possible structures.

5.5 Possible Solution Approaches

5.5.1 Introduction

Three possible solution approaches for the structure subproblem are briefly described. The first approach involves solving the reduced MINLP problem given by Equations 3-9, 3-10, and 5-3 to 5-6, with "standard" MINLP solution algorithms. In particular, the Outer Approximation, Equality Relaxation Method (OA/ER) developed by Kocis and Grossmann (1987), is summarized. The second approach involves converting the MINLP to an integer linear program (ILP). This solution approach requires the solution of a number of ILP's and the use of bounding arguments to get an optimum. The third approach involves the use of an approximate method to obtain near-optimal solutions. The major benefit of this approach is the reduction in computation time required to get a good (although not necessarily optimal) solution.

TABLE 5-4.
NUMBER OF STRUCTURES AS A FUNCTION OF PROBLEM SIZE

Stages	Units	Structures	Stages	Units	Structures
1	1	1	3	3	6
1	2	3	3	4	60
1	3	7	3	5	390
1	4	15	3	6	2,100
1	5	31	3	7	9,786
1	6	63	3	8	43,260
1	7	127	3	9	187,242
1	8	255			
1	9	511	4	4	24
1	10	1,023	4	5	360
			4	6	3,360
			4	7	25,200
2	2	2	4	8	166,824
2	3	12	4	9	1,020,600
2	4	50			
2	5	180	5	5	120
2	6	602	5	6	2,520
2	7	1,932	5	7	31,920
2	8	6,050	5	8	317,520
2	9	18,660	5	9	2,739,240

5.5.2 Outer Approximation, Equality Relaxation Method

The Outer Approximation, Equality Relaxation Method (OA/ER) developed by Kocis and Grossmann (1987, 1989a,b) can be used to solve MINLP problems. The method obtains optimal solutions for problems satisfying certain criteria. These restrictions include the requirement that binary variables appear only linearly in the objective function and the constraints. The reduced MINLP formulated in Section 5.3 for the structure subproblem satisfies these requirements.

The OA/ER method involves alternately solving a series of continuous NLP's and discrete MILP master problems to obtain a final solution. Fixing the binary variables converts the MINLP into an NLP, which can be solved using NLP solution procedures (e.g., successive quadratic programming, generalized reduced gradient methods, etc.) to obtain an upper bound on the optimal solution. Linear approximations for the nonlinear constraints are then constructed at the NLP solution. These linear approximations are included in a MILP master problem to optimize the binary variables. Solving the MILP provides a lower bound on the overall optimum. The new values for the binary variables are used for the next NLP optimization. The NLP and MILP subproblems are solved alternately until the last MILP exceeds the best NLP solution. Integer cuts are required to eliminate previous solutions in the MILP master problems.

The reported advantage of the OA/ER method is that usually only a few time-consuming NLP optimizations are usually required. However, the master problem still requires the solution of an MILP, which generally involves the use of branch and bound procedures in order to obtain a global optimum. As the number of binary variables increases, the computational load for the master problems increases very rapidly.

The OA/ER method can be used to obtain globally optimal solutions for the reduced MINLP formulated for the structure subproblem. Although the NLP subproblem is quite trivial for this particular MINLP, the number of binary variables can be sizeable depending on the number of stages and available units. For problems with a large number of binary variables, the solution time can be high because of the large number of possible combinations which must be pruned during the branch and bound procedures in the MILP master problems.

5.5.3 ILP's with Bounding Arguments

The number of continuous variables in the reduced MINLP is given by Equation 5-7. These continuous variables consist of the total equipment rental charges (E) and the bottleneck stage processing rates (R_{Lp}). At the optimum, minimizing the objective function causes the constraint on the total rental charges (Equation 5-4) to be tight. Thus, this equality constraint can be used to eliminate E. If the bottleneck processing rates are fixed, then the MINLP becomes an integer linear program (ILP). An ILP is easier to solve than an MINLP but still involves solving a combinatorial problem. In order to guarantee that an optimal solution has been found, an enumerative or branch and bound type procedure must be used.

The overall MINLP can be solved by solving a series of ILP's at different values of the bottleneck rates. Bounding arguments can then be used to find a global optimum. For multiple products, searching over P bottleneck rates is not a trivial task. However, for one product, a line search over the one bottleneck production rate coupled with bounding arguments can be used to determine the optimal structure. The only benefit of this ILP approach is that the globally optimal structure can be determined for single product processes without the need to use an MINLP solution technique.

A bounding approach for the case with only one product is described. The bottleneck rate R_L is varied in an outer loop. At each iteration, an ILP problem must be solved. Using Equation 5-4 to substitute for E in Equation 5-3, the objective function for the single product ILP problem is given by:

$$\min \phi = \phi_0 + \frac{A\left(\sum_{j=1}^{J}\sum_{k=1}^{K_j}\sum_{i=1}^{I_j} y_{ijk}\, e_{jk}\right)}{R_L} \tag{5-16}$$

Since ϕ_o is constant, only the equipment usage charges need to be minimized:

$$\min \phi = \frac{A}{R_L}\left(\sum_{j=1}^{J}\sum_{k=1}^{K_j}\sum_{i=1}^{I_j} y_{ijk}\, e_{jk}\right) \tag{5-17}$$

The constraints are given by Equations 3-9, 3-10, and 5-5. All of these constraints are linear in the binary decision variables y_{ijk}. Integer linear programming algorithms essentially perform a branch and bound search to determine the best combination of y_{ijk}'s.

The search over the bottleneck rate R_L is begun by initializing R_L at a low value to ensure that all possible rates are covered. For the first ILP, the bottleneck rate is set equal to the smallest average rate r_{ijk} for any unit k at any stage i. Solving the ILP identifies the set of processing units with the lowest combined total equipment rental rate. The objective function value provided by the ILP solution is usually an over-estimate of the actual objective function value because the bottleneck rate for the "optimal" set of units R_L^* often exceeds the minimum required bottleneck rate specified for the ILP. This larger rate (R_L^*) is used to determine the actual value for the objective function.

The next ILP is generated by updating the value of the bottleneck rate. In general, increasing R_L decreases the objective function. However, the combined total equipment rental rate may be forced to increase because additional units might be necessary to satisfy the higher required production rate. The updating procedure for R_L depends on whether the solution obtained in the current iteration is the best solution obtained so far. If current solution is the overall best so far, then the new value is given by:

$$R_{Lnew} = R_{Lcurr} + \varepsilon \tag{5-18}$$

where ε is a small perturbation. If the current solution is not the overall best, then the new R_L is given by:

$$R_{Lnew} = R_{Lcurr} \frac{\phi_{E\,curr}}{\phi_{E\,best}} \tag{5-19}$$

A series of ILP's are solved for bottleneck rates updated in this fashion. Eventually, a bottleneck rate will be reached that cannot be satisfied even if all available units are used. At this point, the optimal solution is identified as the best solution found during the solution of the ILP's.

This method is not very efficient for determining the overall optimum. Each ILP involves the solution of a potentially large combinatorial problem. However, the process can be sped up considerably if a good solution is in hand initially. Then, the number of ILP's that must be solved during the bounding process and the computational effort

required for each can be significantly reduced. The campaign time horizon puts a lower bound on the value of the bottleneck rate, but this bound may not help much in pruning the number of possible structures. Another drawback to this procedure is that the number of combinatorial problems (ILP's) that must be solved is not known in advance.

An approximate version of this method can be used to get a reasonable estimate of the overall optimum solution for multiple products. The problem is converted into an approximate single product problem by generating a "hypothetical" product that has the characteristics of all the products. This approach was used by Sparrow et al. (1975) in order to quickly generate an approximate solution for the optimal sizing problem. Here, the average production rates for the hypothetical product (r_{ijkH}) are obtained with the following weighting scheme:

$$r_{ijkH} = \frac{\sum_{p=1}^{P} A_p \, r_{ijkp}}{\sum_{p=1}^{P} A_p} \qquad (5\text{-}20)$$

A series of single product ILP's for the hypothetical product are solved during the search over the bottleneck production rate (R_{LH}). After obtaining the "optimal" structure, the actual objective function is determined. As long as the overall horizon constraint is satisfied, this solution represents a good estimate of the overall optimal solution.

5.5.4 Approximate Solution Strategy

Because of the large number of possible structures to be considered, a good approximate method that produces a near-optimal solution for the structure subproblem in a relatively short period of time is desirable. An evolutionary solution strategy has been developed. This method consists of three major parts: (1) a heuristic for generating a good initial structure, (2) an iterative improvement scheme, and (3) a re-start procedure to escape local optima. A local (or neighborhood) search technique is employed as the improvement scheme and serves as the main engine in the solution algorithm. Local search methods are evolutionary, with neighborhoods of the current best structure systematically searched for better solutions. Local search methods have been used previously in both general and batch production scheduling applications (Musier, 1989).

Simulated annealing, which has been receiving attention recently as a solution approach for combinatorial problems (Das et al., 1989; Malone, 1989), is similar in many respects to local search. However, there are a number of key differences. Rather than searching a neighborhood of structures in a systematic order, potential structures are generated randomly from a set of allowable configurations. Also, new solutions are accepted if they are better or "nearly as good" as the current solution. An analog to temperature determines the probability that the method will accept a solution which is worse than the current one. This "nearly as good" feature allows simulated annealing methods to escape local optima. As the temperature is reduced, the method is less likely to accept new structures that increase the objective function.

There are some disadvantages to simulated annealing compared to local search. First, the best temperature schedule must usually be determined through experience. The temperature schedule has important effects on both the quality of the final solution and the computational effort required. Second, the amount of computational effort can be considerably higher than that required by local search procedures that only accept better solutions. The performance of a local search method depends on the choice of search neighborhood selected for the given problem. Although simulated annealing methods could be applied to the structure subproblem, this work has focused on local search methods because computational requirements were expected to be lower than those for simulated annealing. The development of an approximate solution procedure based on local search algorithms for the structure subproblem is described in detail in the next section.

5.6 Development of Approximate Solution Strategy

5.6.1 Initial Structure Generation

The first step in the overall solution approach is the determination of an initial structure. A heuristic method is desired that will generate a good starting point with relatively little computational effort. Two candidate starting procedures are considered. The first method (Pseudo-Random, PR) requires no special insights into the fundamental structure of the problem. The second method (Line Balancing, LB) is based on two general observations about the nature of the problem

The Pseudo-Random starting procedure assigns the units more or less randomly to the stages in the process. This method requires very little computational effort and serves

as a basis of comparison for the Line Balancing starting procedure. The PR method begins by going through each stage of the process. From the pool of unassigned units, a unit is randomly selected and assigned to the current stage. This process is repeated until all the stages have one unit assigned to them. This first pass guarantees that a feasible structure is generated. The second pass assigns the remaining units. Each unit is considered in turn at this point. A location for each remaining unit is selected randomly from all the possible stage locations of the appropriate task type and the equipment inventory. For example, if there were two possible stage locations for unit k, there would be a one in three chance that the unit would be assigned to neither stage and simply left in inventory.

The quality of the initial structures generated by this method is expected to be quite variable. One possible way to improve the quality of the method would be to implement it a number of times and take the best structure. An important factor in determining how many initial structures should be randomly generated will be the correlation (if any) between the quality of the initial structure and the performance of the local search procedure that follows.

The Line Balancing heuristic is intended to be an improvement on the PR method in terms of the quality of the initial structure. This improvement is based on insights into the nature of the particular problem being dealt with in this work. As discussed previously, the effective production rate of a multi-stage process is limited by a bottleneck stage. The location of the bottleneck stage depends on the assignments of the available units to the various process stages. Two general observations that stem from the bottleneck concept can be used to develop an improved method for determining an initial structure:

(1) A good structure has few units that are vastly under-utilized, i.e. the line is reasonably well balanced.
(2) The number of distinct limiting (bottleneck) average production rates for the process is much less than the total number of structures.

The main point of the first observation should make intuitively good sense. The truth of the second observation can be seen by considering an example.

Suppose six units are available for a two stage process. Assume each unit has an average rate of 50 for the first stage and an average rate of 75 for the second stage. If only one unit is placed at stage 1, the bottleneck rate for the process will be 50 regardless of whether one, two, three, four, or five units are used at stage 2. In this case, the average

rate of 50 corresponds to five different structures. Clearly, the best structure of these five would have only one unit at stage 2. The second observation implies that searching over rates could be more effective than searching over structures. The difficulty is that the values of the bottleneck rate for a process and a given set of available units are also discrete and not readily obvious. However, these two concepts can be used to construct a method that finds a good initial structure relatively quickly.

The Line Balancing procedure is summarized in Figure 5-6 and outlined below. The current "target rate" is the bottleneck production rate that every stage must satisfy. A range of target rates is considered because it is not known in advance what the best bottleneck rate will be. The first step in the procedure is to determine an appropriate range of target rates to search. In order to reduce the total number of structures to consider, initial structures with only one or two units at a stage are considered. Merged equipment usage charges and rates are calculated for all possible pairs of units. Then, for each target rate, the unit or pair of units with the lowest usage charge that satisfies the current target rate is assigned to the stage where its rate is maximum. This assignment process is repeated until all stages are filled. Then, the overall procedure is repeated for all the target rates. The best structure obtained is used as the initial structure. The number of target rates to try is specified by the user. Generally, five target rate values have been used in this study.

5.6.2 Local Search Procedure

A Local Search (LS) technique is employed as the improvement strategy. From the initial structure or the current best structure, a neighborhood of structures similar to the current structure is searched in some systematic fashion (Papadimitriou and Steiglitz, 1982). The search neighborhoods used for this work are the 1-change neighborhood and a restricted version of the 2-change neighborhood. Recall that the process being carried out here is the assignment of the available equipment units to the stages in the process. A 1-change is the move of a single unit from one location to another. A location is either a stage in the process or the equipment inventory. A general 2-change involves selecting any two units and assigning them to any combination of two locations. The restricted 2-change considered here consists of a single pairwise switch of two units, where the units involved simply exchange locations. This restricted neighborhood greatly reduces the computational effort required by the method (Musier, 1989).

When a proposed move or switch results in an equal or better (lower cost) structure, the change is immediately accepted. When no additional moves or switches can

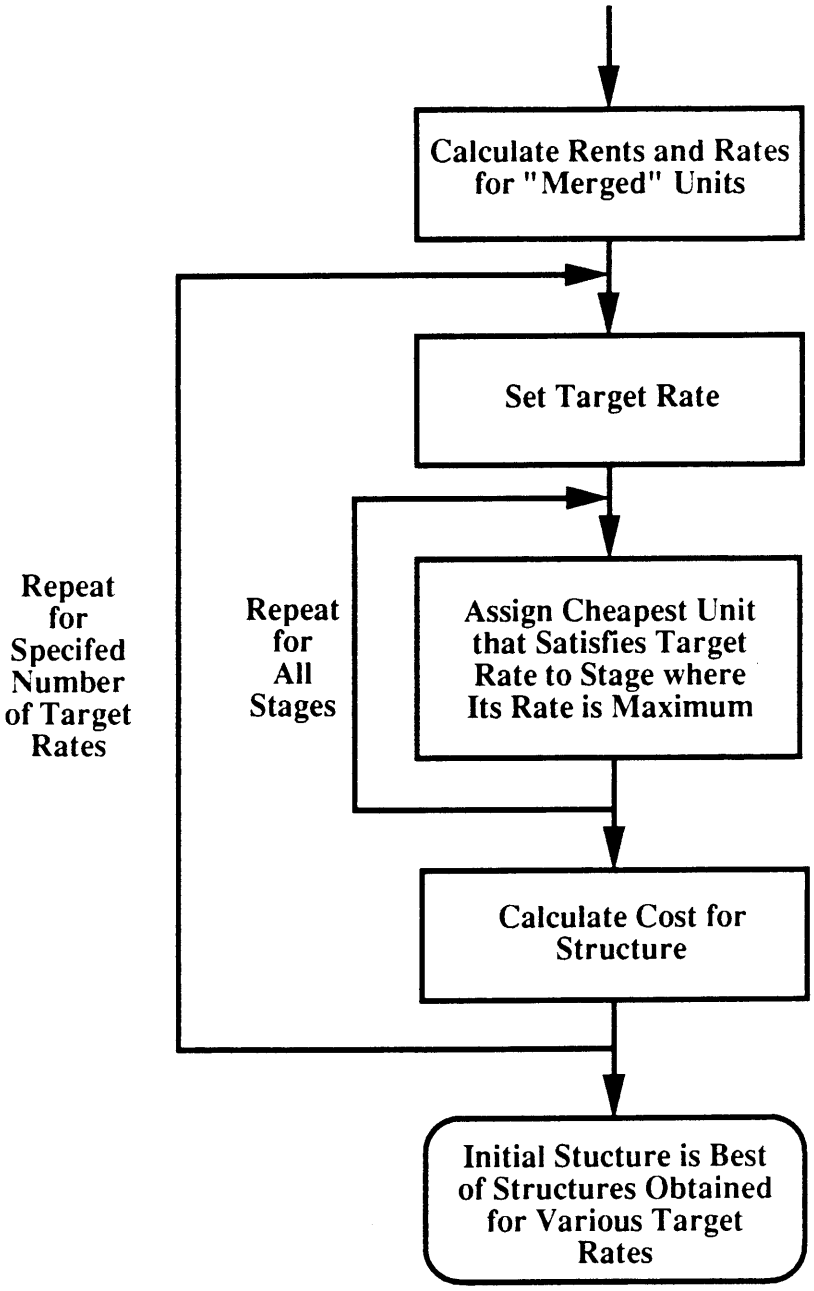


FIGURE 5-6. LINE BALANCING INITIAL STRUCTURE GENERATION METHOD

be made that will reduce the value of the objective function, then that structure is said to be both 1-optimal and "2"-optimal (in our restricted sense). If this local optimum structure were also the global optimum, then it would be n-optimal. In other words, no n-change could be made that would reduce the objective function value. However, an n-change neighborhood includes all possible repositionings of n units and thus contains all possible structures. The n-change neighborhood would require complete enumeration to search. Since an n-optimal structure must also be both 1-optimal and "2"-optimal, the hope is that a structure that satisfies these necessary (but not sufficient) conditions will be close to the optimal solution. In addition, the number of structure evaluations needed to find a structure that satisfies these weaker necessary conditions is generally much less than the total number of structures.

The search is carried out in the following manner (see Figure 5-7). Equipment units are ordered from smallest to largest within each type on the basis of average rate over all possible stages. The search procedure begins by attempting single moves. Starting with the smallest unit, moves to inventory and then to each other stage location are tried. In order to maintain a feasible structure, a unit is not moved if it is the only unit currently assigned to a process stage. The procedure moves to the next unit in turn after the first move that equals or lowers the objective function value. Types of units are considered in the order specified by the user at the start of the program.

After cycling through all the units and attempting moves, the procedure goes on to try pairwise switches. Starting with the smallest unit, switches are tried with progressively larger units. As soon as an equal or better structure is obtained, the method moves on to the next larger unit and tries switches with it. Again, the sets of units for each type are considered in the user specified order. After completing this set of switches, the algorithm returns to the single move phase. Rounds of moves and switches are carried out until no changes are accepted during a consecutive set of moves and switches. The resulting structure satisfies the 1-optimal and restricted "2"-optimal necessary conditions.

One of the peculiarities to this particular problem formulation is that multiple structures can have the same objective function value. This situation might occur if one stage severely limited the bottleneck rate for the entire process. It would be conceivable that the two cheapest units for another task type could both exceed the bottleneck rate for two different stage locations. Both permutations would be equally good. To prevent endless cycling, the search procedure terminates if two full passes of moves and switches do not improve the objective function, even if changes are accepted.

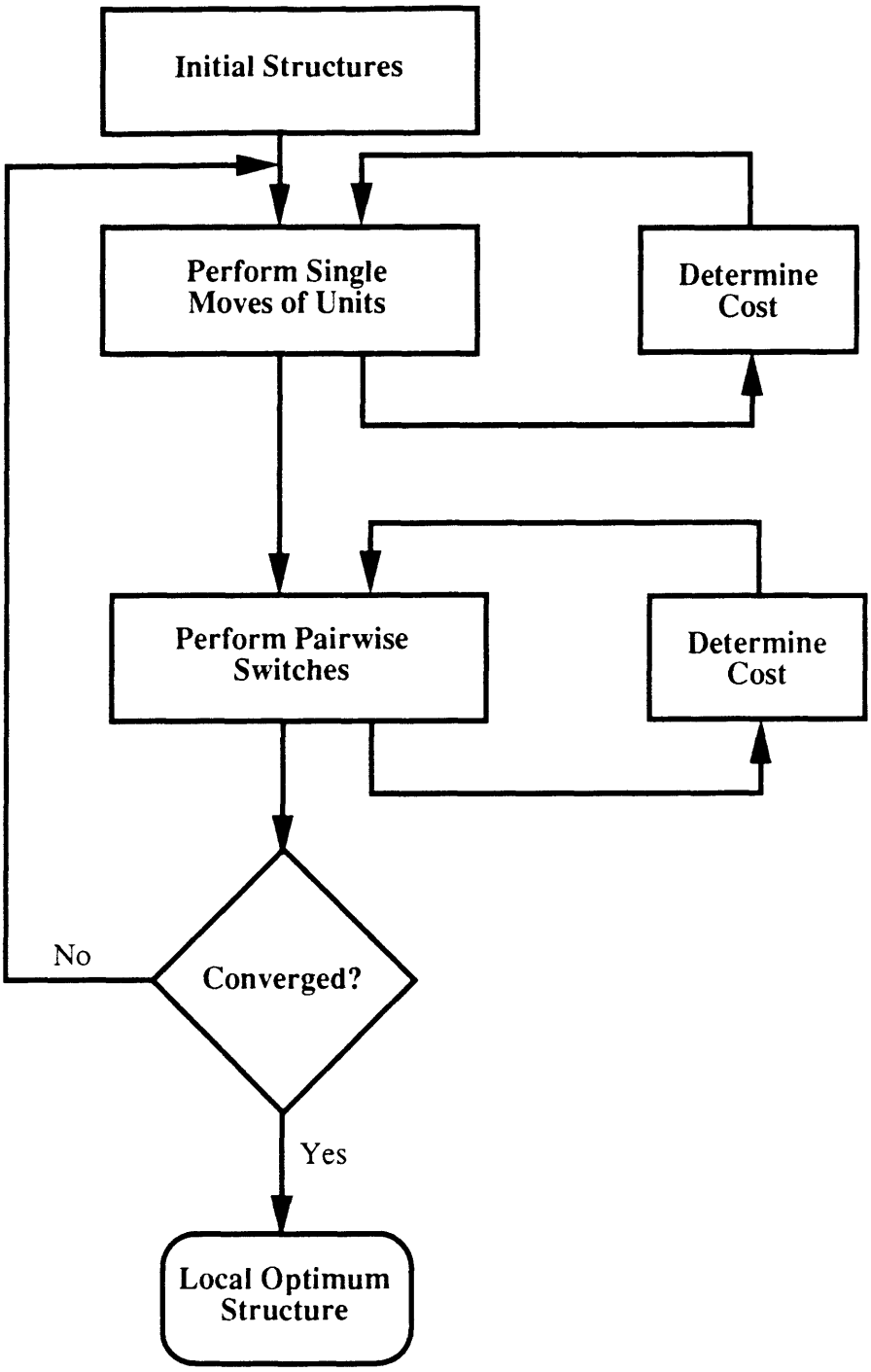


FIGURE 5-7. LOCAL SEARCH PROCEDURE

5.6.3 Re-Start Procedure

One typical problem with local search techniques such as this one is the possibility that the algorithm will get stuck in a local optimum far from the global optimum. The procedure may not be able to jump out of a local optimum because the neighborhood is too small. Consider the following example. The LS procedure has terminated with structure A shown in Figure 5-8. However, it is known that structure B is the global optimum. With the 1-change and restricted "2"-change neighborhoods that the algorithm is using, two steps are required to get from structure A to structure B. In other words, structures A and B are not in the same 1-change and "2"-change neighborhoods. Since the intermediate structures have objective function values greater than that of structure A, structure B cannot be reached.

In order to avoid having the algorithm terminate at sub-optimal structures for the reason given above, a re-start procedure (RS) has been developed. This re-start method attempts to generate structures "similar" to the local optimum but outside the original 1-change and "2"-change neighborhoods. After generating a new structure, the LS procedure is begun again. The final solution is taken as the best of the local search solutions after one round of re-starts.

The new structures for re-start are generated in a two step process. First, for the unit types that did not contain the overall bottleneck stage for the process, the stage locations of the units are permuted in order to maximize the limiting stage rate for that type only. Since the same units are being used in the structure and the bottleneck rate for the overall process remains the same, the objective function values for each of these permuted structures is the same. The goal is to make the non-bottleneck types as efficient as possible. The stage locations for the units of the type that does contain the overall bottleneck stage in the original converged local search solution are permuted in the second step. All permutations of stages for the bottleneck type are generated in conjunction with the more efficient structures obtained for the non-bottleneck types, and the LS procedure is repeated for each new structure. Going back to the example in Figure 5-8, the re-start procedure would immediately generate structure B, which is the optimal structure.

The number of re-starts grows factorially with the number of stages of the bottleneck type. The extra computational load may be judged excessive for the improvement in algorithm performance. Rather than use permutations of stages, pairwise

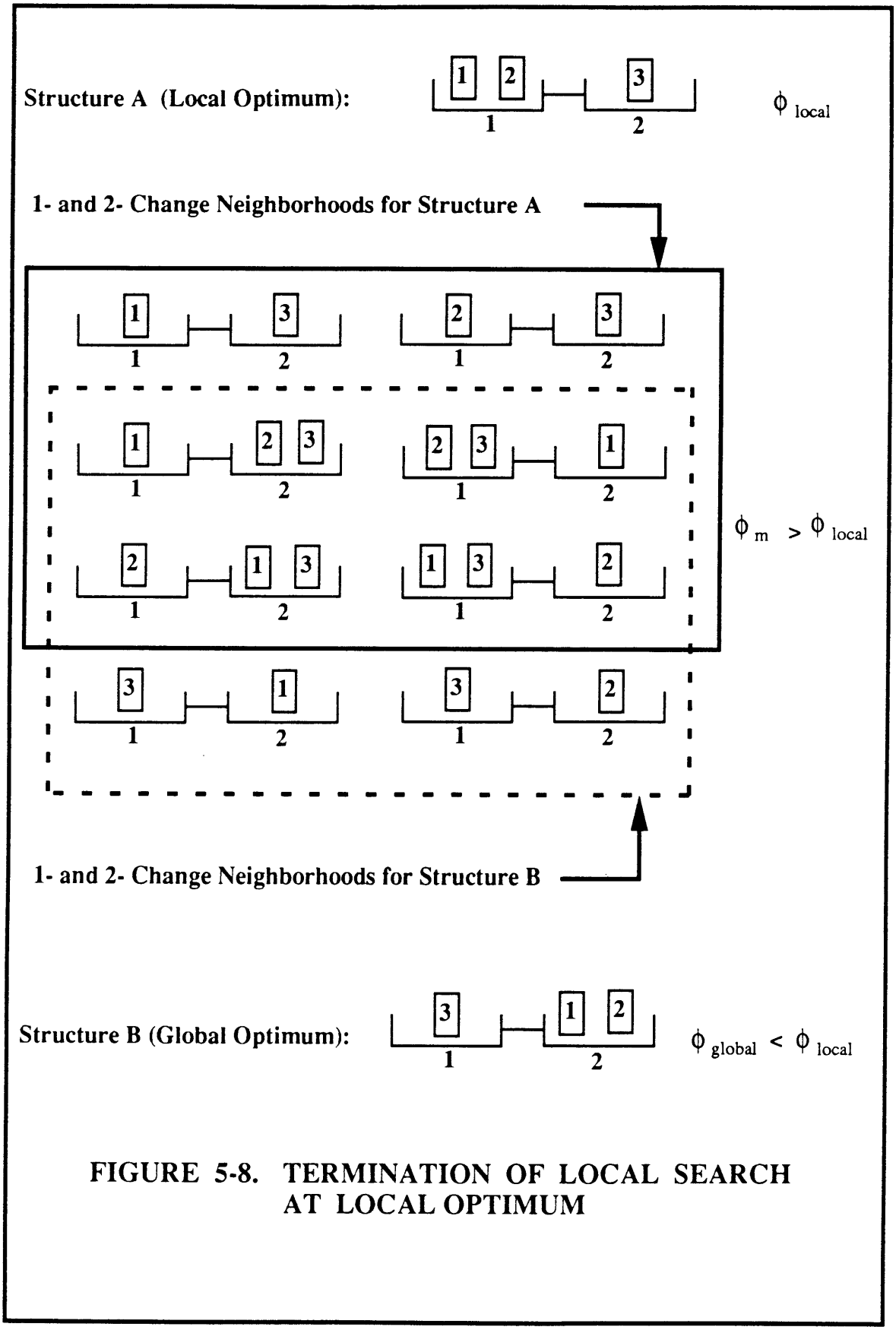


FIGURE 5-8. TERMINATION OF LOCAL SEARCH AT LOCAL OPTIMUM

switches and circular rotations could be used to reduce the computational load while still providing an opportunity to escape poor local optima.

5.7 Test Problems

5.7.1 Overview

A series of test problems is carried out to determine the performance of the approximate solution strategy described above, particularly in regard to its ability to obtain near-optimal solutions in reasonable computation times. Six "variations" of the solution procedure are studied: (1) pseudo-random initial structure method, PR, (2) line balancing initial structure method, LB, (3) PR with local search, PR/LS, (4) LB with local search, LB/LS, (5) PR/LS with re-start, PR/LS/RS, and (6) LB/LS with re-start, LB/LS/RS.

Three types of test problems are used to examine the relative and absolute performance of the six variations listed above. First, a simple two stage process is examined to make a preliminary evaluation of the performance of the algorithm in an absolute sense. The relative performance of the initial structure generation methods and their effects on the the local search procedure are also considered. Next, a battery of random problems allows the relative performance of the six "methods" to be considered as a function of a number of problem parameters. Finally, the globally optimal solutions for a few randomly generated problems are obtained by using the ILP procedure described in Section 5.5.3. For these problems, absolute performance measures can be considered. In the following sections, each of these test problems and the key results obtained are described and discussed.

5.7.2 Two Stage Problems

The first test problem is a simplified two stage process like the ones described previously in Section 4.5. The process consists of a batch reaction followed by a batch distillation. With three different sized reactors and six different columns available, a total of 441 distinct equipment structures are possible. Changing the reaction time or temperature affects the performance (size factors) of the process, and thus the relative rankings of the possible equipment structures. The costs for all 441 possible structures are enumerated for four different sets of conditions.

For each of these four sets of conditions, the structure subproblem is solved using the first four variations of the solution procedure. Because there is only one stage of each

type, the re-start procedure does not generate any new structures to test. The results are summarized in Table 5-5. The ranking and scaled cost are shown for the best structure generated by each method. Costs are scaled by the global optimum, so a value of 1.00 is the best possible.

The results of this small test problem indicate that the proposed approximate algorithm does have some promise. The LB heuristic clearly outperforms the PR method for all the example problems. Increasing the number of starting points would certainly improve the performance of the PR method, but the superiority of the LB method is so large that the extra computational effort required to give comparable results would be excessive. The LS method significantly improves on the initial structures provided by both the PR and LB methods. However, after applying the LS procedure for these four cases for both types of initial structures, the much better LB starting points yield only slightly better final results. The LB/LS results are quite good, with the worst of the four cases resulting in a structure within five percent of the global optimum cost. Also, two of the four problems are solved optimally by the LB/LS method.

5.7.3 Random Test Problems: Relative Performance

Although the results for the two stage problems are good, only four cases have been examined, and each case contains a relatively small number of alternatives. In order to make a better evaluation of this approximate solution strategy, larger problems must be tried. Therefore, a set of 1500 random test problems are solved to examine the relative performance of the six variations listed above over a range of problem parameters. Because the number of possible structures for these problems is generally too large to conveniently enumerate, the results on each problem are scaled by the best solution found by any of the six methods. These "relative" problems include cases where the number of possible structures ranges from 10^3 to 10^{18}.

The data for the processing rates and equipment usage charges are randomly generated according to the following scheme. First, three factors (S_{jk}, E_{jk}, R_{ij}) are randomly selected from given uniform distributions. The average processing rate for unit k of type j at stage i (r_{ijk}) and the equipment usage charge for unit k of type j (e_{jk}) are given by the following expressions:

$$r_{ijk} = R_{ij}S_{jk} \tag{5-21}$$

TABLE 5-5.
RESULTS FOR 2-STAGE PROBLEMS

Reaction Conditions		Solution Method				
Rxn Time	Temperature	PR	PR/LS	LB	LB/LS	
40 hr	345 K	1.268	1.025	1.032	1.01	Cost*
		250	10	16	5	Rank**
45 hr	345 K	1.203	1.050	1.069	1.046	Cost
		180	30	52	25	Rank
50 hr	345 K	2.706	1.035	1.049	1.000	Cost
		412	12	25	1	Rank
75 hr	337 K	1.500	1.088	1.123	1.000	Cost
		219	18	37	1	Rank
Average Results		1.669	1.049	1.068	1.014	Cost
		265	17.5	32.5	8.0	Rank

* Costs are scaled by the optimum cost. Values of 1.000 indicate optimal solution.

** Rank out of 441 structures that are possible for this problem

$$e_{jk} = e_{jk_0} + E_{jk} (S_{jk})^{0.6} \qquad (5\text{-}22)$$

This scheme causes trends in average rate over a range of stages to be the same for all units in a given problem. Likewise, trends in average rate over a set of units are the same for all stages. For example, it is reasonable to expect that a big reactor should have a higher production rate than a smaller reactor for all reaction stages. In addition, the use of a 0.6 power law term means that in general bigger units cost more to use, although this need not always be the case.

Sets of runs to determine the effects of the number of types of units (J), the number of processing stages (I_j), and the number of units available (K_j) on the relative performance of the approximate methods are carried out. The three problem parameters are varied one at a time, keeping the other two values constant. The "nominal" values for these parameters for this study are J = 3, I_j = 3, and K_j = 6. I_j and K_j values are kept the same for each type j. For each run with a given set of parameter values, the six methods (PR, LB, PR/LS, LB/LS, PR/LS/RS, LB/LS/RS) are compared based on results averaged over 100 random problems. Performance measures used to compare the six methods include the quality of solution (cost scaled by the cost of the best performing method), the percentage of problems for which a method obtains the best solution, the computation time, and the number of structures evaluated.

The results for the relative performance for the LB and PR initial structure generation methods are shown in Figure 5-9 for a number of sets of problem parameter values. The objective values are scaled by the objective function value of the best performing method of the six listed above. The LB method again does much better than the PR method. The relative superiority of the LB method over the PR method generally increases as the number of possible structures increases. In addition, the variability of the PR results is quite high (as one would expect) and increases with increasing problem size. Although the performance of the LB method does degrade slightly as the number of stages approaches the number of units available, it is clear that the LB method greatly outperforms the PR method over a wide range of problem parameter values. Again, the number of starting points required for the PR method to match the performance of the LB method would be computationally excessive.

When the LS procedure is performed, the final structures are significantly better than the initial structures generated by the PR and LB methods. The extent of the improvement for the LB starting point is shown in Figure 5-10. The LS procedure

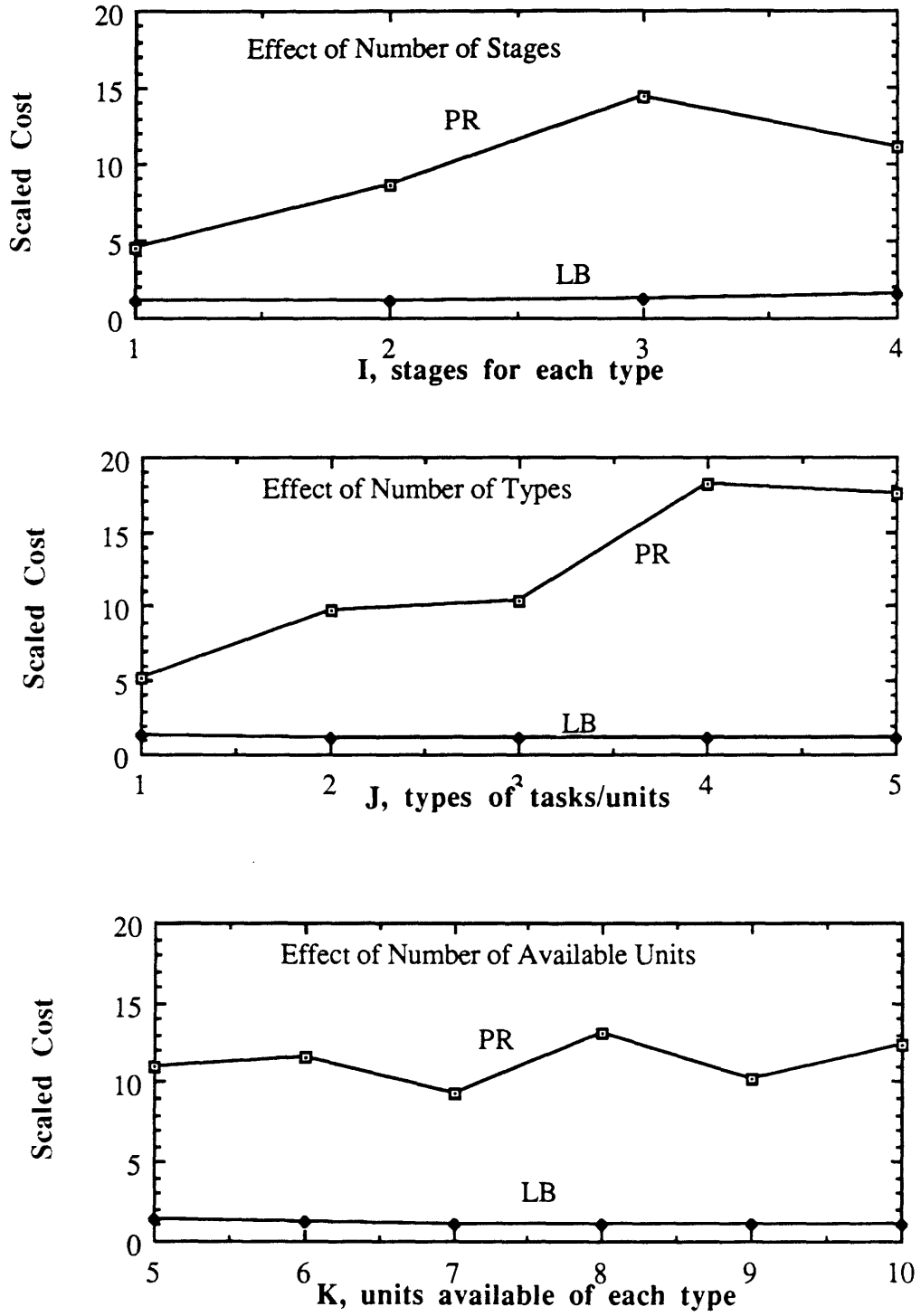


FIGURE 5-9. COMPARISON OF PR AND LB METHODS

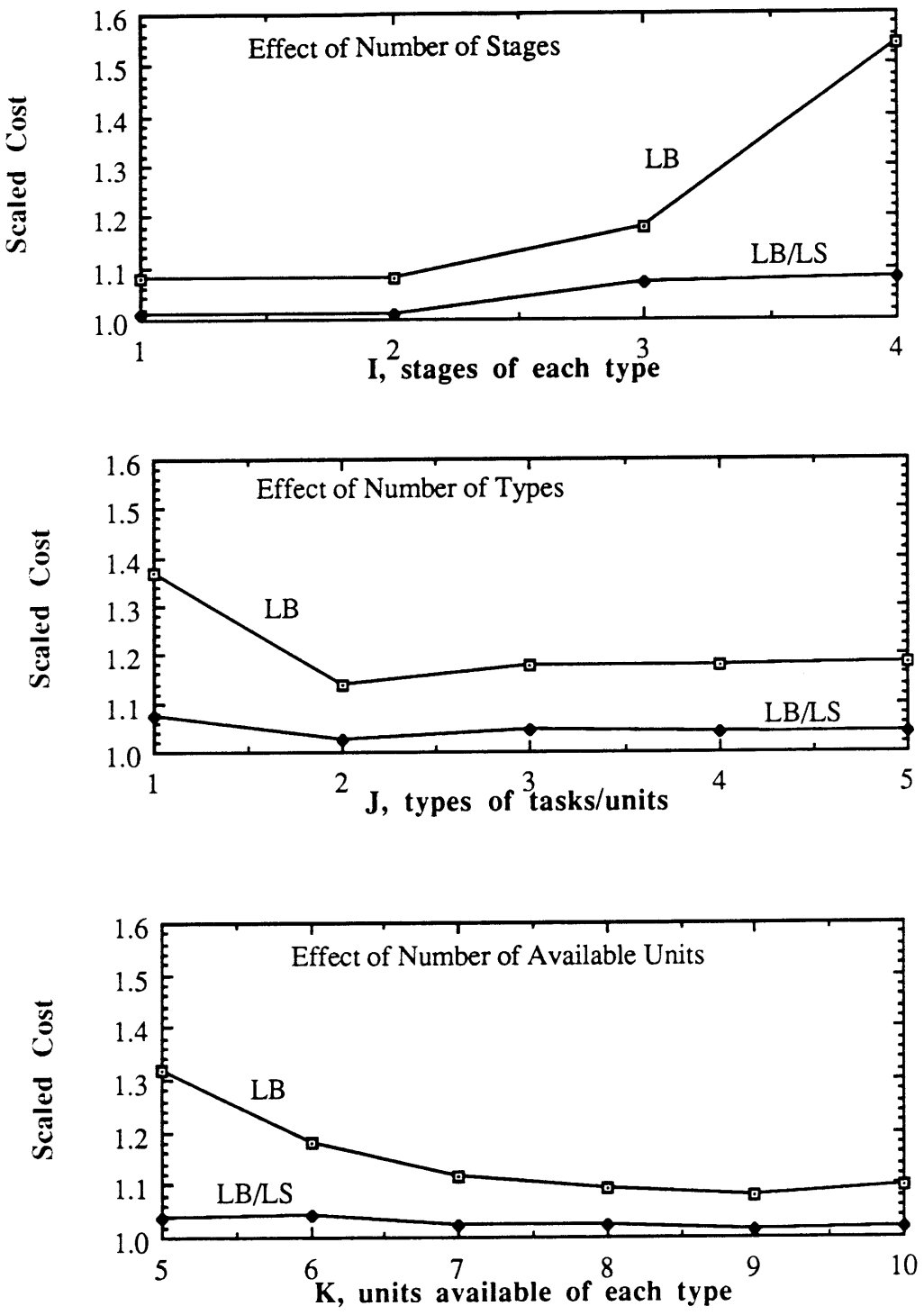


FIGURE 5-10. COMPARISON OF LB AND LB/LS METHODS

typically obtains cost reductions in the range of 15 percent over the initial LB solutions. The means for these two methods are statistically different based on the standard deviations for the 100 randomly generated test problems. Even greater relative improvements are found in the PR case because the initial PR structures are generally so poor.

A comparison of the results for the PR/LS and LB/LS methods provides some interesting insights on how the choice of initial structure generation method affects the quality of the final solution obtained. Figure 5-11 shows that the LB/LS method outperforms the PR/LS method by about 10 percent over a range of J types and K_j units. However, the results are less clear-cut over the range of I_j stages examined. Again, for cases when the number of stages approaches the number of units available, the LB-based method shows some degradation in performance. One very interesting result is that despite the large differences in the quality of the initial structures generated by the PR and LB methods, the final results after LS are quite close. This point would indicate that the quality of the initial solution is not of paramount importance to the LS method. However, some slight benefits appear to be gained from the much better starting points in the form of solutions that are approximately 10 percent better on the average.

A second interesting result concerns the relative performance of the LB and PR/LS methods. (See Figures 5-10 and 5-11.) These two methods give similar quality results for a number of problem parameter values. Although the PR/LS method is more likely to find the best structure than the LB method, the fact that these two are even close shows how well the LB method performs. Since the LB method is based on simple heuristics, its computational requirements are much less than those for the PR/LS method, especially for larger problems. The fact that the two methods have similar average performance (within 5 percent) while the PR/LS method finds the best structure significantly more often might mean that the PR/LS procedure occasionally "gets stuck" in a relatively poor local optimum. These "bad outings" would increase the average cost result for the PR/LS. Since the LB/LS is starting "much closer" to its final destination, the relatively high quality of the LB starting point means that even if the LB/LS method gets stuck, it will not be left with an extremely poor solution. This hypothesis might explain the better performance of the LB/LS method over that of the PR/LS method.

After the local search method finds a 1-optimal and "2"-optimal structure, the re-start procedure described previously generates additional structures with which to begin the LS procedure again. Therefore, the PR/LS/RS and LB/LS/RS methods will always report solutions at least as good as those of the corresponding LS methods. Figure 5-12 shows

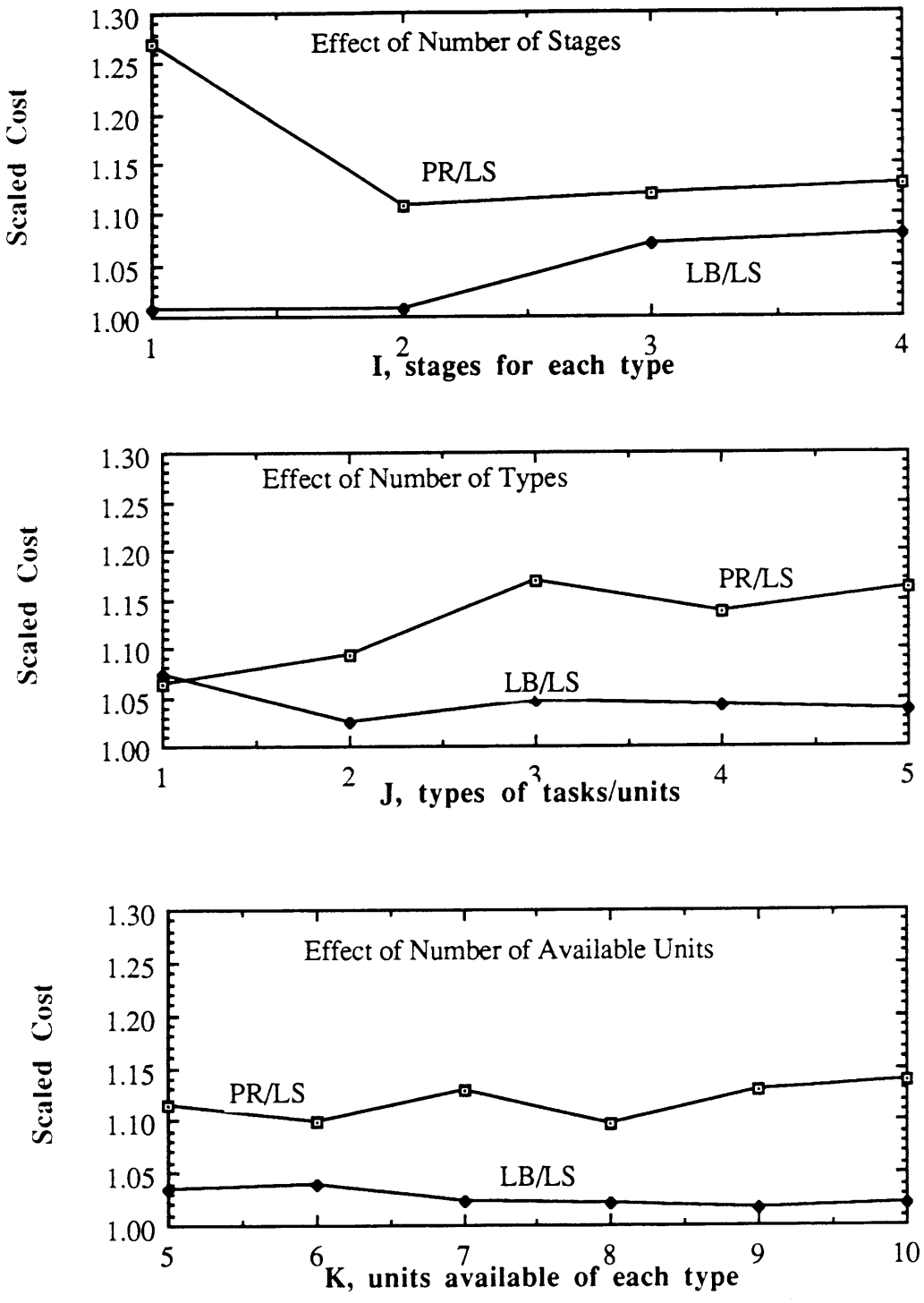


FIGURE 5-11. COMPARISON OF PR/LS AND LB/LS METHODS

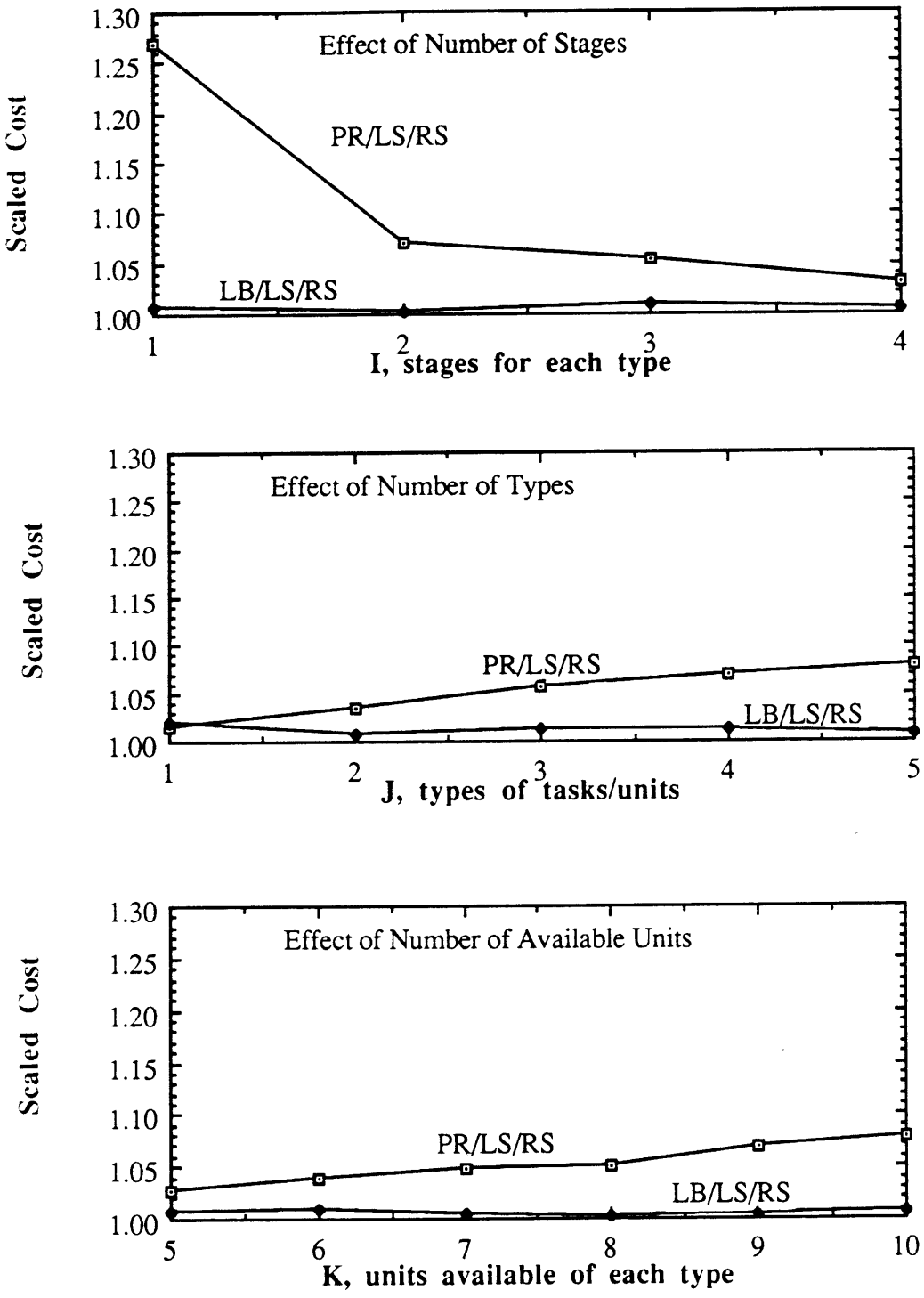


FIGURE 5-12. COMPARISON OF RE-START METHODS

the results for the two re-start methods. In general, the LB-based method performs slightly better than the PR-based method again. The margin of difference narrows somewhat to approximately 5 to 7 percent. The point for $I_j = 1$ is simply a comparison of the PR/LS and LB/LS results because the re-start procedure only works when there are two stages or more. The LB/LS/RS method finds the best structure of those found by the six variations of the algorithm over 85 percent of the time on average. Therefore, the average scaled results for the LB/LS/RS method are essentially 1.0 for almost all the parameter values tested.

The use of the re-start procedure provides an average improvement of about 3 percent over the LB/LS method and about 18 percent over the LB method for the 1500 random test problems solved. This improvement in the solution quality comes at the expense of added computational effort. Figure 5-13 compares the cpu time requirements for the LB, LB/LS, and LB/LS/RS methods. The results for the PR-based methods are similar. The re-start method takes the most time, followed by the local search method. The computational loads for these methods go up with increasing problem size. These time increases are reasonable with the exception of the LB/LS/RS method for cases when the number of stages increases. As described earlier, the re-start method generates all permutations of the current unit to stage assignments. Naturally, the computational effort for this procedure will increase factorially with increases in the number of stages. Using pairwise switches and circular rotations rather than all permutations would make the computational time more reasonable for large problems while still providing the chance to escape local optima.

In general, the computational loads for all six of these methods are quite low. The random test problems require cpu times on the order of seconds for solution on a DEC Microvax II. Another encouraging result in terms of algorithm performance is shown in Figure 5-14. The fraction of the total number of structures that are evaluated by the local search method is plotted as a function of problem size on a log-log plot. The fraction evaluated decreases substantially with increasing problem size. This plot indicates that the number of structure evaluations increases with the number of possible structures to the 0.07 power. The real benefit of using good approximate techniques is that much better solutions to large problems can be obtained without the need to do an almost infinite number of evaluations in order to obtain a good near-optimal solution.

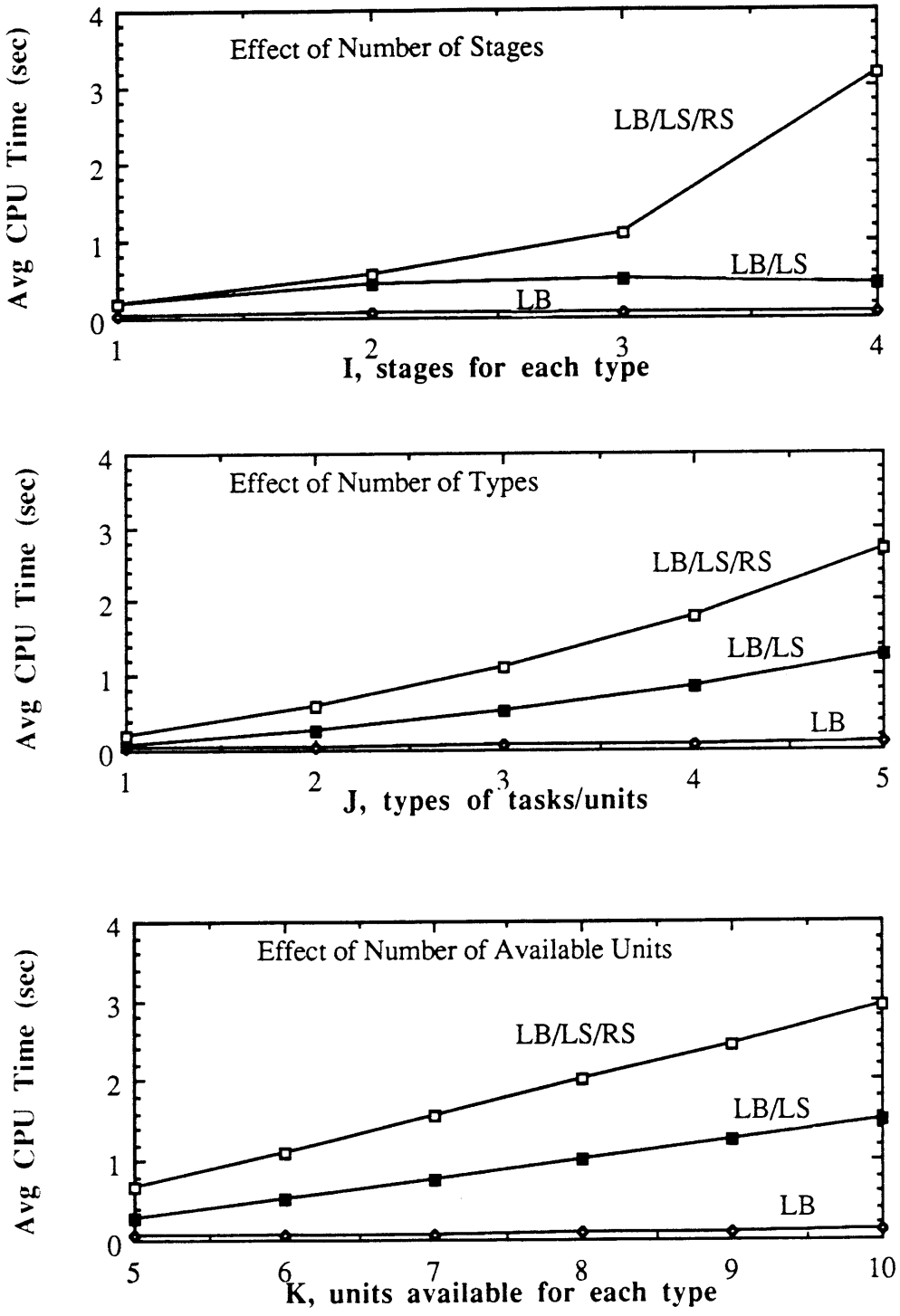


FIGURE 5-13. COMPARISON OF CPU TIMES FOR LB-BASED METHODS

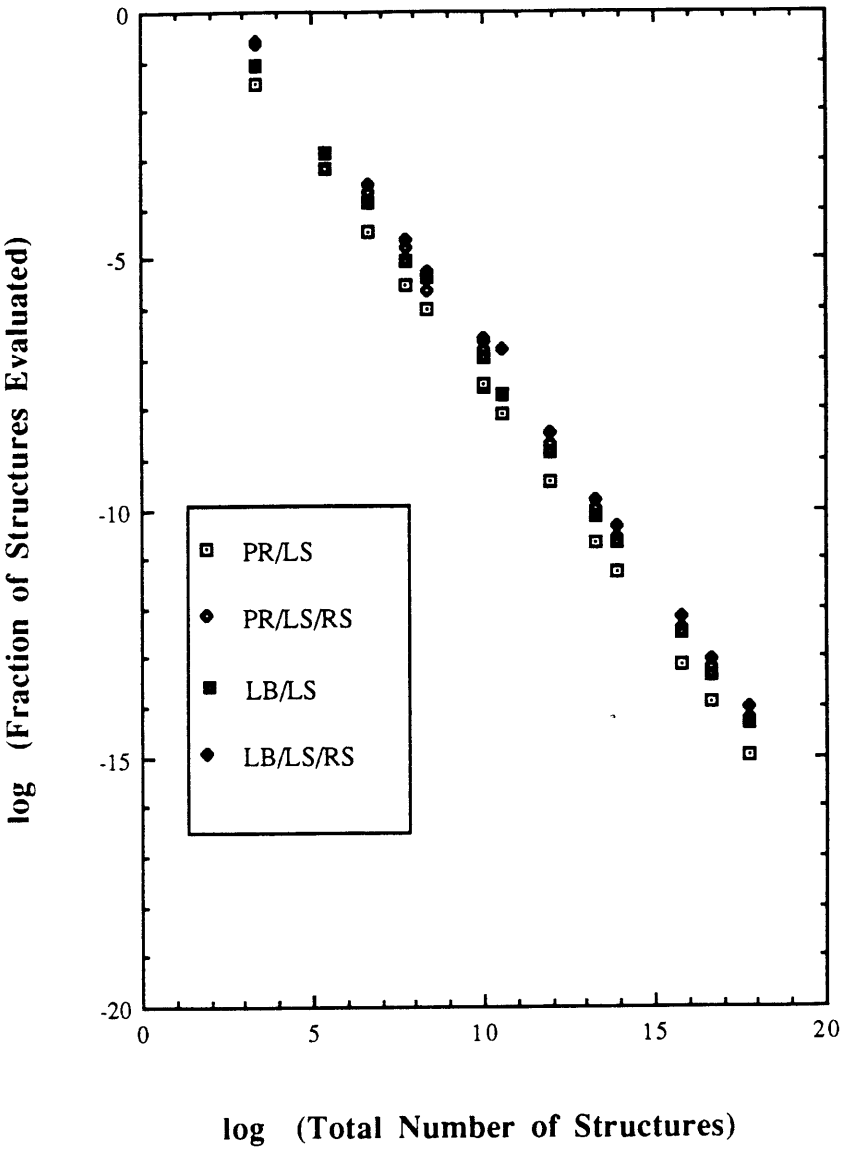


FIGURE 5-14. FRACTION OF STRUCTURES EVALUATED BY LOCAL SEARCH METHOD AS A FUNCTION OF THE NUMBER OF POSSIBLE STRUCTURES

5.7.4 Random Test Problems: Absolute Performance

The results for the random test problems discussed in the previous section provide insights into the relative performance of the six variations of the solution algorithm developed. However, the question remains as to whether any of the six variations is succeeding in coming up with near-optimal solutions. For the small two stage problems discussed previously, all possible structures could be enumerated to determine the absolute performance of the algorithm. For more interesting, industrial sized problems, enumeration of all possible structures to find the global optimum is not practical.

Seven problems with randomly generated problem data are solved using the ILP solution technique described in Section 5.5.3 in order to find the global optimum structure. The number of possible structures for these problems ranges from 2500 up to 81 trillion. Each of these problems is solved using the six approximate methods in order to determine their absolute performance. Table 5-6 shows the results and the comparison with the global optimum. The approximate algorithm developed in this work does indeed obtain near-optimal results for all seven cases. In five of the seven cases, the LB/LS/RS method obtains the global optimum. In addition, the PR/LS and the LB/LS methods get the optimum in three and two cases, respectively. At least one method finds the optimum in each of the seven problems.

The computational requirements on a DEC Microvax II are on the order of cpu seconds (0.1 to 5) for all six variations for each of the seven problems. The number of ILP's required for these problems ranged from five to fifteen. Although the ILP's were not accurately clocked, each one took anywhere from tens of seconds to a couple hours of cpu time, depending on the problem size. Thus, these approximate methods perform very well in an absolute sense in just a fraction of the time required by "exact" methods.

5.7.5 Summary of Overall Results

The performance of the new algorithm on the seven problems just described is quite impressive. However, seven problems does not constitute a large sample for statistical purposes. Unfortunately, the effort required to find the optimal solutions for these problems is so large that solving enough problems to get a solid statistical sample is impractical. However, the fact that at least one of the methods finds the optimal solution for each of the seven problems is fairly convincing.

TABLE 5-6.
RESULTS FOR RANDOM PROBLEMS WHICH HAVE BEEN SOLVED OPTIMALLY USING ILP METHOD

Problem Characteristics				Scaled Costs for 6 Variations					
J	I_j	K_j	# Structures	PR	PR/LS	PR/LS/RS	LB	LB/LS	LB/LS/RS
2	2,2	4,4	2500	4.71	1.120	1.078	1.00	1.000	1.000
2	3,3	6,6	4.4E6	7.16	1.079	1.077	1.39	1.050	1.000
2	3,3	6,6	4.4E6	4.47	1.000	1.000	1.05	1.000	1.000
3	3,3,3	4,4,4	2.2E5	1.44	1.000	1.000	1.16	1.083	1.083
3	3,3,3	6,6,6	9.3E9	4.98	1.006	1.006	1.23	1.006	1.000
3	3,3,3	8,8,8	8.1E13	2.65	1.177	1.130	1.09	1.039	1.000
5	1,...,1	6,...,6	9.9E8	3.10	1.000	1.000	1.20	1.090	1.090
Average Performance				4.07	1.055	1.042	1.16	1.038	1.025

** Costs scaled by Global Optimum Cost. Value of 1.000 indicates optimal solution.

Additional support for the conclusion that the local search methods are obtaining near-optimal solutions is provided by Table 5-7, which compares the average results for the three types of test problems. There is especially good agreement between the two sets using the randomly generated problems. Extrapolating back over the random problems, the relative results should provide estimates within a few percent for the performance of the six methods with respect to the optimal solutions. The following conclusions are drawn from the results of the three sets of problems:

(1) The approximate solution strategy based on local search obtains near-optimal structures within approximately 3 to 4 percent of the global optimum with very modest computational requirements.

(2) The Line Balancing heuristic method obtains initial structures that are within approximately 20 percent of the global optimum on the average.

(3) The Local Search procedure is affected only slightly by the quality of the initial structure.

(4) The Re-Start procedure generates only marginal improvements in most cases but requires a relatively high computational load compared to the other elements of the approximate solution strategy.

5.7.6 Areas for Additional Work

The algorithm could be fine tuned in a number of ways to improve its performance and efficiency. These efforts at fine tuning have not been emphasized as a part of this study, but one possible adjustment is discussed here. As noted previously, the LB-based methods seem to have more difficulties when the number of stages approaches the number of units. In fact, this degradation in performance is caused by poor selection of the target rates.

Figure 5-15 shows a plot of the minimum cost as a function of the bottleneck rate for one of the two stage problems. Each point represents the low cost structure for the given bottleneck rate. This plot shows the discontinuous, non-unimodal nature of the objective function. Also, the 441 possible structures represent only 32 distinct bottleneck rates. These low cost solutions are spread over a range of rates. When the number of stages approaches the number of units, this range of rates is much narrower, often causing

TABLE 5-7.
SUMMARY OF RESULTS OF TEST PROBLEMS FOR APPROXIMATE SOLUTION PROCEDURES BASED ON LOCAL SEARCH METHOD

Problem Type	Solution Method					
	PR	PR/LS	PR/LS/RS	LB	LB/LS	LB/LS/RS
Relative (Avg of 1500)	11.15	1.130	1.066	1.187	1.036	1.008
2-Stage (Avg of 4)	1.67	1.049		1.068	1.014	
Random (Avg of 7)	4.07	1.055	1.042	1.160	1.038	1.025

Problem Type	General Description
Relative	Randomly generated problems. Results scaled by the best solution found on each problem to show relative performance of methods. 10^3 to 10^{18} possible structures
2-Stage	Simple 2-Stage problems. Results scaled by the optimum for each problem. 441 possible structures
Random	Randomly generated problems. Results scaled by the optimal solution for each problem. 10^3 to 10^{13} possible structures

Solution Method	General Description
PR	Pseudo-Random method for generating initial structure
PR/LS	Local Search scheme applied to PR initial structure
PR/LS/RS	Re-Start method applied to final PR/LS structure
LB	Line Balancing method for generating initial structure
LB/LS	Local Search scheme applied to LB initial structure
LB/LS/RS	Re-Start method applied to final LB/LS structure

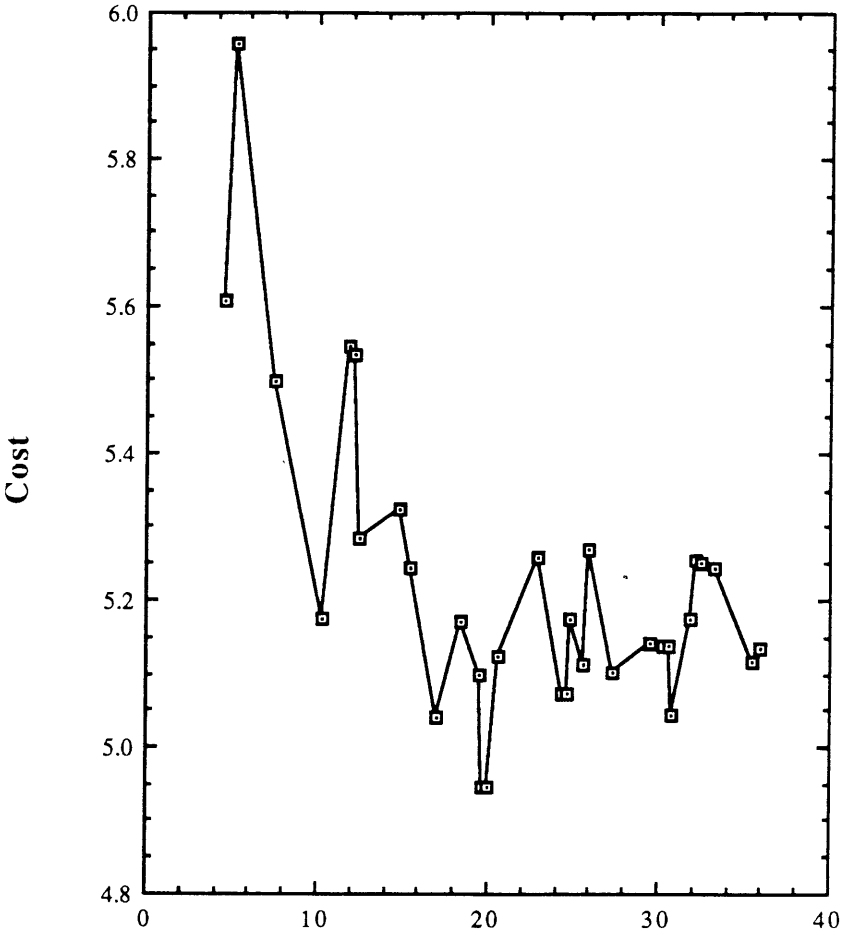


FIGURE 5-15. COST AS A FUNCTION OF BOTTLENECK RATE

the LB method to try only one feasible target rate. Thus, the first target rate tried at the low end gives a poor solution while the second and subsequent target rates exceed the maximum possible bottleneck rate. Adjusting the way that target rates are selected should improve the performance of the LB-based methods.

There is a very attractive feature of algorithms based on local search techniques. Note that the local search routine simply manipulates the assignments of the units to stages. The routine makes no assumptions about the form of the objective function in going about its work. As long as providing the structure allows the objective function to be calculated in a straight-forward way, the LS procedure should work fine. However, although significant changes in the form of the objective function can be easily incorporated, the relative performance of the LS algorithm with respect to the optimal solutions would need to be re-evaluated for any new class of problems.

A number of changes in the problem formulation, which could be accomplished by simply changing the subroutine that calculates the objective function, might be considered. For example, costs based on the number of batches could be added. Also, the assumption of unlimited intermediate storage could be changed. A policy of no intermediate storage could be used as long as the appropriate logic for determining the average production rate is coded into the objective function evaluation routine. The local search approach appears promising as a relatively fast, general purpose solution procedure for large combinatorial problems dealing with the structure of batch processes. These structural issues include the selection of units from an inventory of existing units, the number of parallel units at each stage, and the location of intermediate storage. The use of local search methods for batch production scheduling applications has already been noted.

5.8 Extension to Multiproduct Operation

The reduced MINLP for the structure subproblem formulated in Section 5.3 concerns the design of a multiproduct process to produce P products. However, only single product test problems have been done in Section 5.7 to assess the performance of the approximate solution strategy. This section describes how the approximate methods can easily be extended to handle the multiproduct case.

For the PR-based methods, multiple products require only changes in the way the objective function is calculated for a given structure. No changes to the solution method itself are required. On each iteration, a subroutine calculates the overall cost by summing

up the contributions of the various products. Since the number of possible structures is not affected by the number of products, the performance of the local search procedure should be similar to that obtained for the single product case.

For the LB-based methods, the multiple products case requires some minor adjustments in the LB procedure used to get an initial structure. Since there are P products, the LB should search over P target rates to get a good initial structure. A multivariable search is avoided by following the lead of Sparrow et al. (1975) and generating a "hypothetical" product. (See discussion in Section 5.5.3.) Using rate information for the hypothetical product, a single product LB procedure is carried out. The LS method is then used for the multiproduct case with an appropriate subroutine to calculate the full objective function.

Another possible alternative is the use of the data for the hypothetical product as the basis for the LS procedure as well. The only potential advantage to this approach would come in cases when the objective function evaluation requires significant computational effort. In such cases, some computational savings would be obtained because only one (rather then P) cost evaluation would be required for each structure.

Although the approximate methods should be easily extended to multiple products, the performance of these methods in terms of solution accuracy has not yet been verified on test problems. Increasing the number of products complicates the MINLP problem by increasing the number of continuous variables. Because the computational effort required to obtain the global optimum for problems of this type is so large, these test problems have been left for future work.

5.9 Chapter Summary

This chapter has focused on the structure subproblem, which involves determining the best unit to stage assignments for cases when the process performance is fixed. Because existing equipment units are used, the optimization variables for the problem are discrete. This combinatorial optimization problem has been formulated as an MINLP. Although standard solution techniques can be used to obtain globally optimal solutions, the potentially large number of discrete alternatives can result in high computation times for solution. An approximate method based on local search techniques has been developed that obtains near-optimal solutions with very little computational effort. The performance of the solution approach has been characterized on a number of test problems. Because the

computational effort required for these approximate methods grows relatively slowly with problem size, these techniques can be applied to large problems without the need for excessive computation time. Extensions of the solution approach to cases with multiple products have also been discussed.

Chapter 6

THE GENERAL PROBLEM: PERFORMANCE AND STRUCTURE INTERACTIONS

6.1 Introduction

This chapter addresses the more general problem described in Chapter 3. The effects of both process performance and equipment assignment are now considered together. Although the combined optimal design problem can be formulated as a large MINLP, a decomposition strategy has been pursued in this thesis for reasons given in Section 3.4. Chapters 4 and 5 have discussed solution approaches for the two subproblems generated by partitioning the decision variables into a set of continuous performance variables and a set of discrete structure assignment variables. This chapter investigates ways to get the solution procedures for the two subproblems to work together in order to generate solutions to the overall problem.

The magnitudes of any interactions between the two groups of decisions must be determined in order to develop an effective solution procedure for the combined problem. Significant interactions would require that a solution approach be developed that coordinates the solution of both subproblems in such a way that an optimal solution is obtained for the overall optimization problem. However, a lack of interactions would mean that the performance and structure subproblems could be completed in a single pass, sequential fashion, using the solution strategies already discussed in previous chapters.

This chapter begins by determining the types (and magnitudes) of interactions that occur between the process performance variables and the equipment to stage assignments. The performance subproblem for a simple two stage example problem from Chapter 4 is solved for all possible structures. Possible solution approaches are then discussed, focusing on two alternative nesting arrangements of the performance and structure decisions. Two example problems are solved to illustrate the potential merits of the two methods. Finally, the key issues for future work on the development of better algorithms for the combined problem are identified and discussed.

6.2 Performance / Structure Interactions

Interactions between the two groups of decisions are expected because the unit sizes and rates affect the calculation of the average production rates for the stages during the performance subproblem. In a sense the choices of equipment units for a stage can be thought of as process intensity decisions. If a stage is the bottleneck, more units can be assigned (i.e., "more intense operation") to increase the processing rate although at the expense of increased rental charges. Changing the location of the bottleneck stage is also expected to affect the way the performance load is distributed among the stages through the values of the process operating variables and times.

The existence of significant interactions between performance and structure is demonstrated by solving the performance subproblems for all possible structures for a simple two stage example problem from Chapter 4. The two stage process with temperature dependent kinetics and a perfect splits column from Example Problem #3 has been selected for this study. The relatively simple process model and small number of possible structures (only 441) make complete enumeration of the performance subproblems feasible in a reasonable amount of computation time. The performance subproblem for this example involves determining the reaction time and temperature that minimizes the total operating cost.

Table 6-1 shows the available equipment inventory for this example problem. There are 7 possible ways to assign the three available reactors to stage 1. Likewise, there are 63 possible ways to assign the six columns to stage 2. Even though the columns come in only three pot sizes, none of the units is identical to any other because column operation depends on both the pot size and the vapor rate. Because the reactor and column assignments can be made independently, there are a total of 441 (7 x 63) possible structures for this process.

The two main decision variables for the performance subproblem are the reaction temperature and operation time. Setting these values determines the final reaction compositions, which in turn set the column performance because the perfect splits column model is being used. Figure 6-1 shows the optimal reaction temperature plotted against the optimal reaction time for each of the 441 possible structures. Clearly, the optimal conditions vary significantly from structure to structure.

TABLE 6-1.
AVAILABLE EQUIPMENT UNITS FOR 2-STAGE PROBLEM

Unit Type (Number Available)	Unit Volume (Liters)	Vapor Rate (mol/hr)	Usage Charge ($/hr)
REACTORS (3)	1000	-	30
	2000	-	45
	4000	-	75
DISTILLATION COLUMNS (6)	500	800	30
	500	1600	55
	1000	1200	50
	1000	2400	75
	2000	3200	140
	2000	4800	200

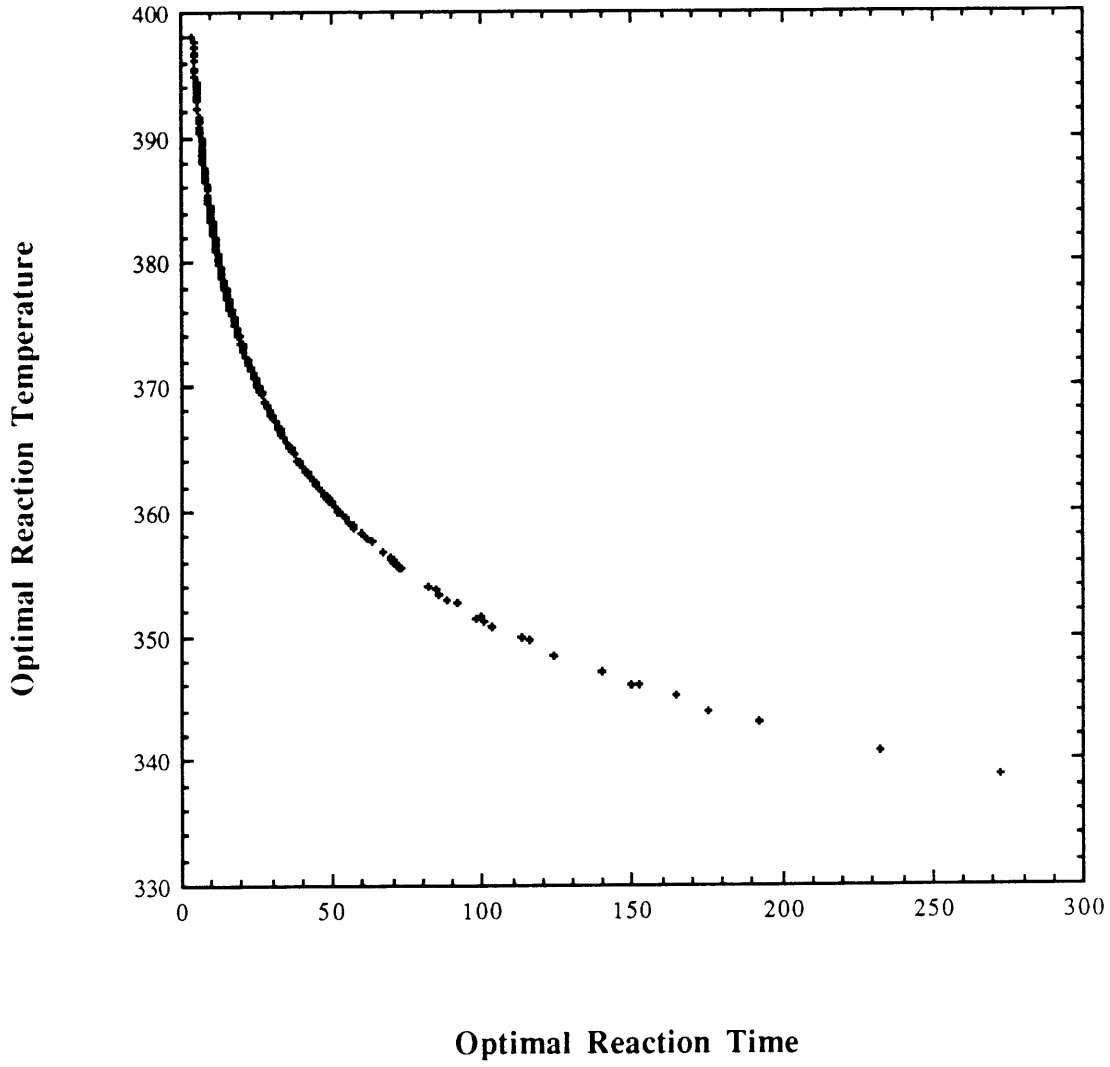


FIGURE 6-1. OPTIMAL TEMPERATURES AND REACTION TIMES FOR ALL 441 POSSIBLE STRUCTURES

The optimal reaction conditions shift to take into account the relative processing "speeds" of the two stages. In Figure 6-2, the optimal reaction temperature and optimal reaction time are both plotted against M, which is an approximate measure of the ratio of the potential average rates of the two stages. Higher values of M indicate that the column is the "slow" stage, while lower values correspond to the reactor as the slow stage. For low values of M ("slow" reactor), Figure 6-2 shows that the optimal conditions are high temperature and short reaction time in order to speed up the reaction step. At high values of M (column is "slow"), the opposite results are obtained. Lower reaction temperatures increase the maximum yield of product B, so fewer batches are required. Figure 6-3 shows the final composition of B at the end of the reaction for each of the 441 structures plotted against M. This plot confirms the shift towards higher conversion to B that occurs when the column becomes the limiting stage.

The interactions between process performance and structure are confirmed by plotting the optimal cost as a function of the optimal reaction time for the 441 possible structures. Ideally, the cost should be plotted as a "surface" of points as a function of optimal reaction time and temperature. Since Figure 6-1 shows that the optimal temperature is apparently some type of function of the optimal reaction time, a simpler 2-dimensional representation is possible. Figure 6-4 shows that the optimal cost varies quite significantly over the range of possible structures. However, most of the low cost structures have optimal reaction times in the range of 30 to 60 hours. These better structures also tend to occur when the two stages have approximately similar processing capabilities, as shown in Figure 6-5.

The results of the 441 performance subproblems indicate that varying the process structure significantly affects the optimal total cost and the optimal process operating conditions. There is clearly a best structure that should have its performance optimized. Thus, performance and structure considerations must both be taken into account in order to get the overall optimum solution.

6.3 Overall Solution Approach

6.3.1 Decomposition Strategy

A decomposition strategy is proposed for solving the overall problem in which process performance and structure must be optimized together. The use of a decomposition strategy allows the methods developed for the two subproblems to be considered for use in

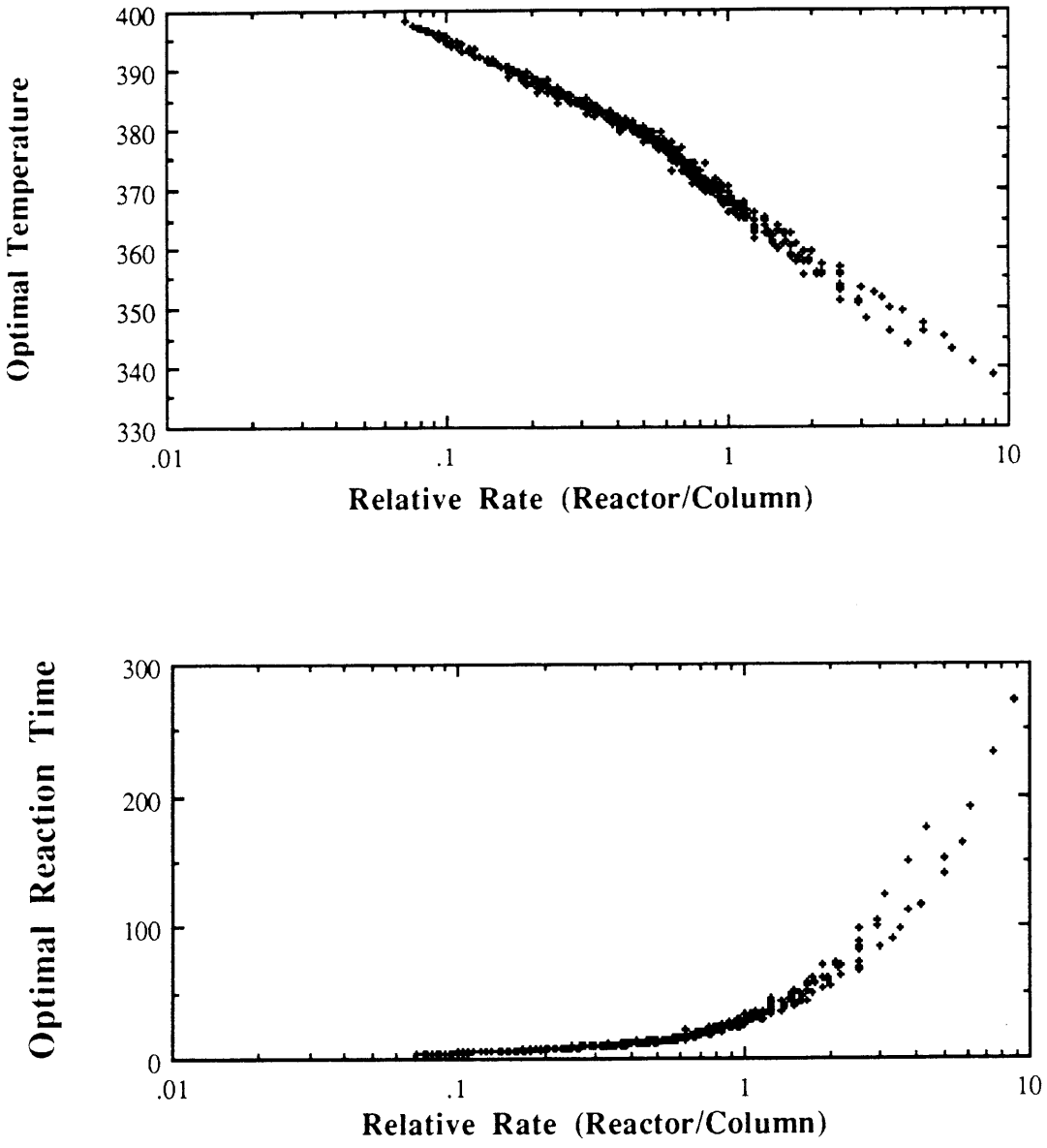


FIGURE 6-2. EFFECT OF THE RELATIVE PROCESSING RATE ON THE OPTIMAL PROCESSING CONDITIONS FOR 441 POSSIBLE STRUCTURES

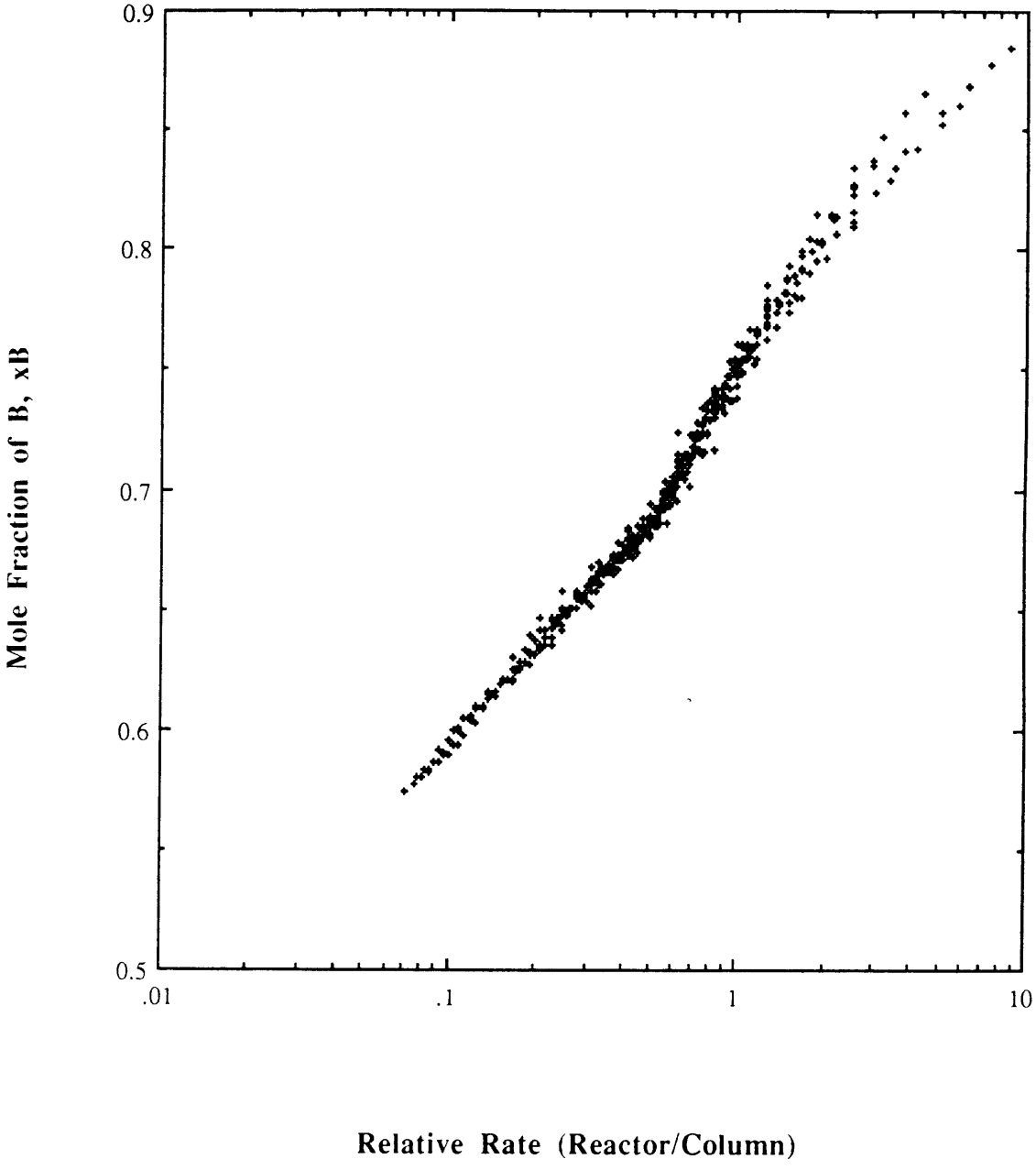


FIGURE 6-3. EFFECT OF THE RELATIVE PROCESSING RATE ON THE PERFORMANCE LOAD DISTRIBUTION

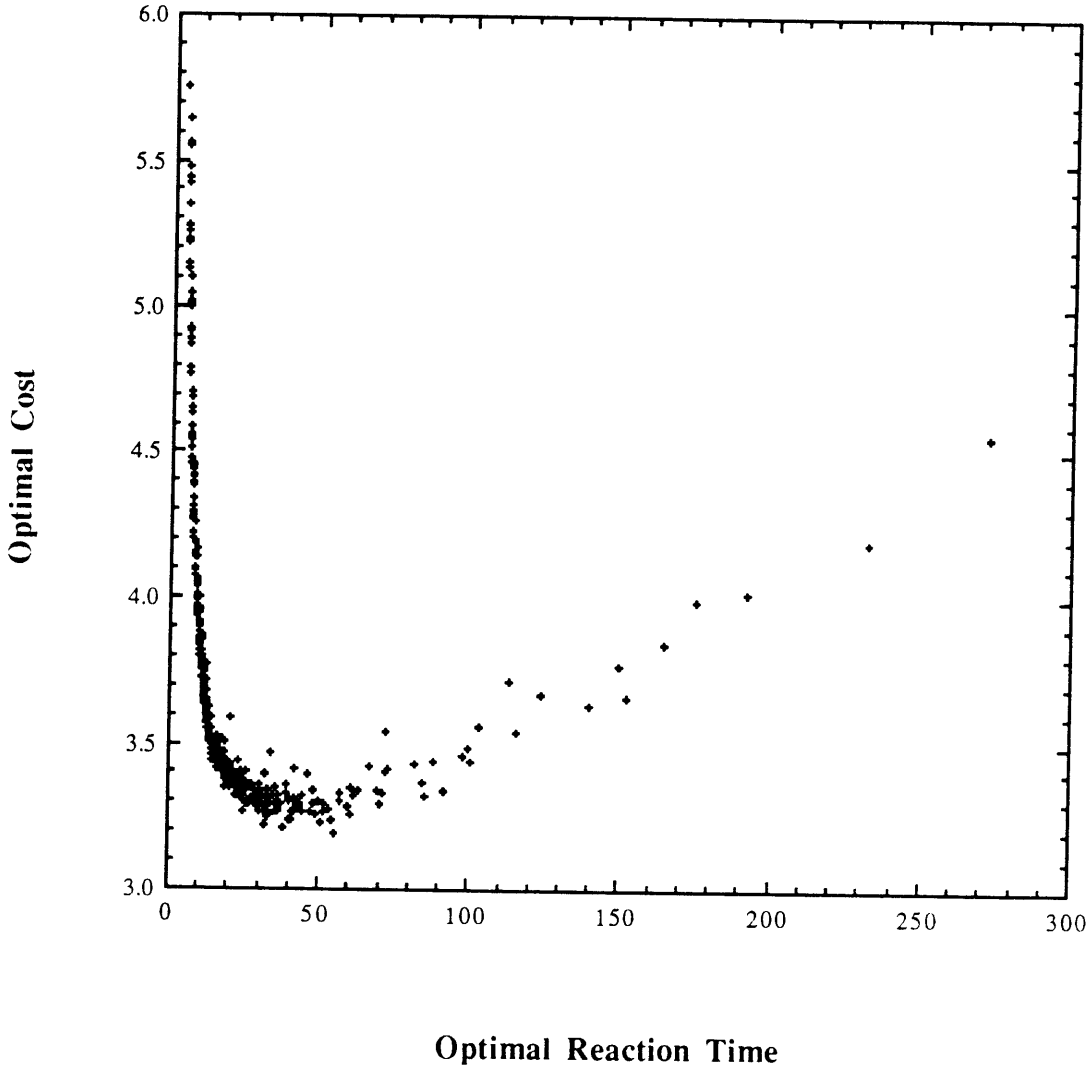


FIGURE 6-4. OPTIMAL COSTS FOR 441 POSSIBLE STRUCTURES

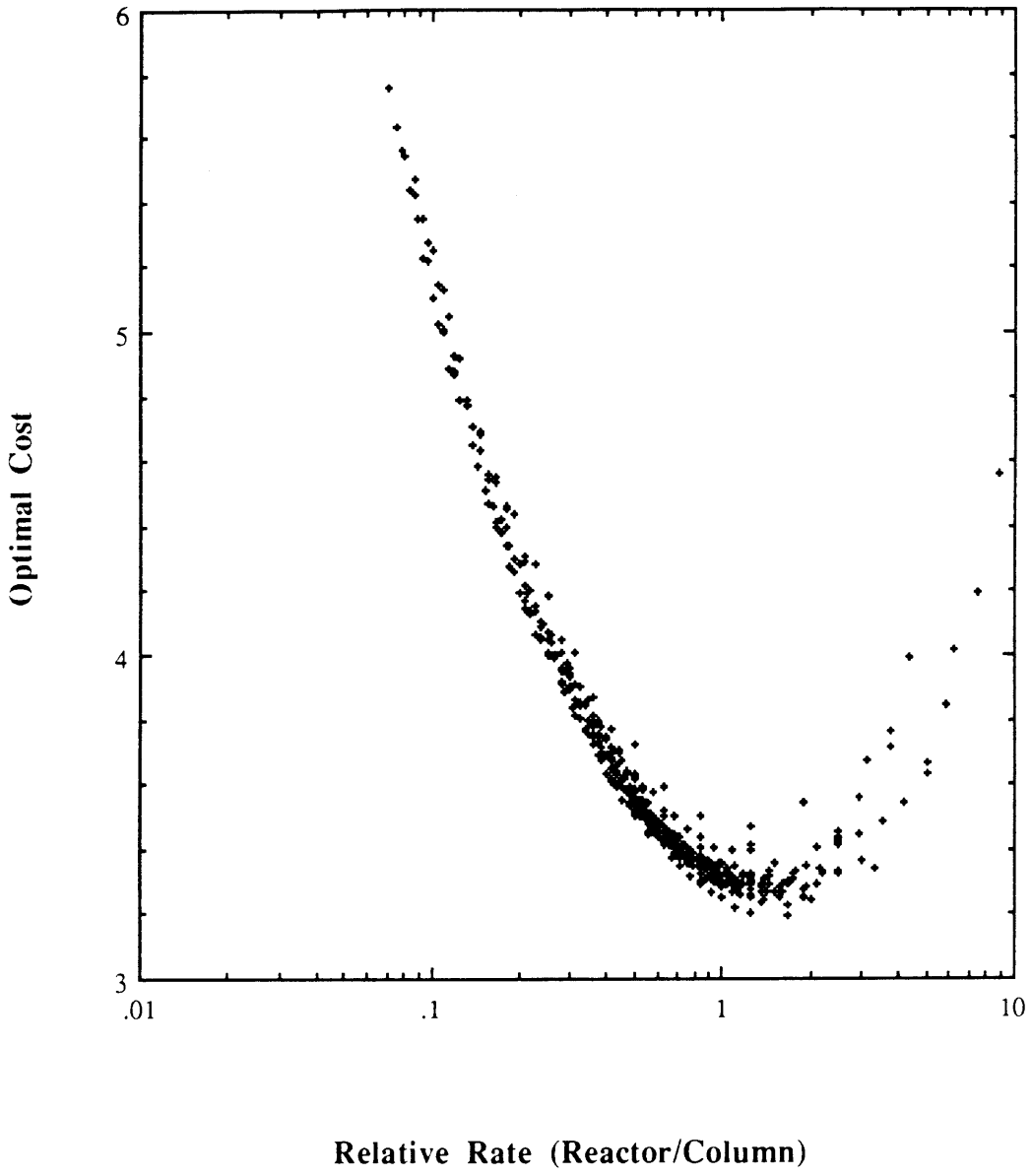


FIGURE 6-5. EFFECT OF RELATIVE PROCESSING RATE ON OPTIMAL COST FOR 441 POSSIBLE STRUCTURES

the combined problem. Also, the knowledge and experience obtained by studying the two subproblems individually can then be applied to the combined problem. However, the issue of how the two subproblems should be coordinated must be resolved.

Two alternative nesting strategies are proposed as possible candidate solution approaches for the overall problem. As shown in Figure 6-6, each method essentially consists of solving a series of nested subproblems, with the difference between the two coming in the nesting arrangement. Placing the continuous performance subproblem inside the discrete structure subproblem is one alternative, which shall be referred to as the SOPI method (Structure Outside, Performance Inside). The other arrangement (POSI method; Performance Outside, Structure Inside) has the structure subproblem embedded inside the performance subproblem.

Three issues are important in evaluating the attractiveness of these two embedding schemes. The first consideration is the availability and quality of methods for the subproblems in the particular nesting arrangement. Second, the average quality of the final solution obtained, i.e., its closeness to the global optimum, is another key concern. The final issue involves the amount of computation time required. Ideally, the best nesting arrangement would use well established solution techniques for both the inner and outer optimization problems and obtain the globally optimal solution in very little computational time. Naturally, this ideal situation is not likely. Since the problem is a large combinatorial one, there will probably be a trade-off involving the computational effort and the quality of the final solution. The nature of this trade-off, particularly with respect to changes in problem size, is therefore a key issue in determining which embedding strategy is preferred.

6.3.2 SOPI Nesting Arrangement

The SOPI nesting arrangement places the performance subproblem in the inner loop, inside the structure optimization. Good methods are available to handle both the performance and structure subproblems for this particular arrangement. The SOPI embedding scheme is conceptually pleasing because the continuous optimization problem is solved inside the combinatorial problem. The performance subproblem is an NLP which can be solved to obtain an exact optimum in many cases (at least in theory). With the potentially large combinatorial problem in the outer loop, approximate methods can be used reduce the computational effort required to obtain a near-optimal structure. The evaluation

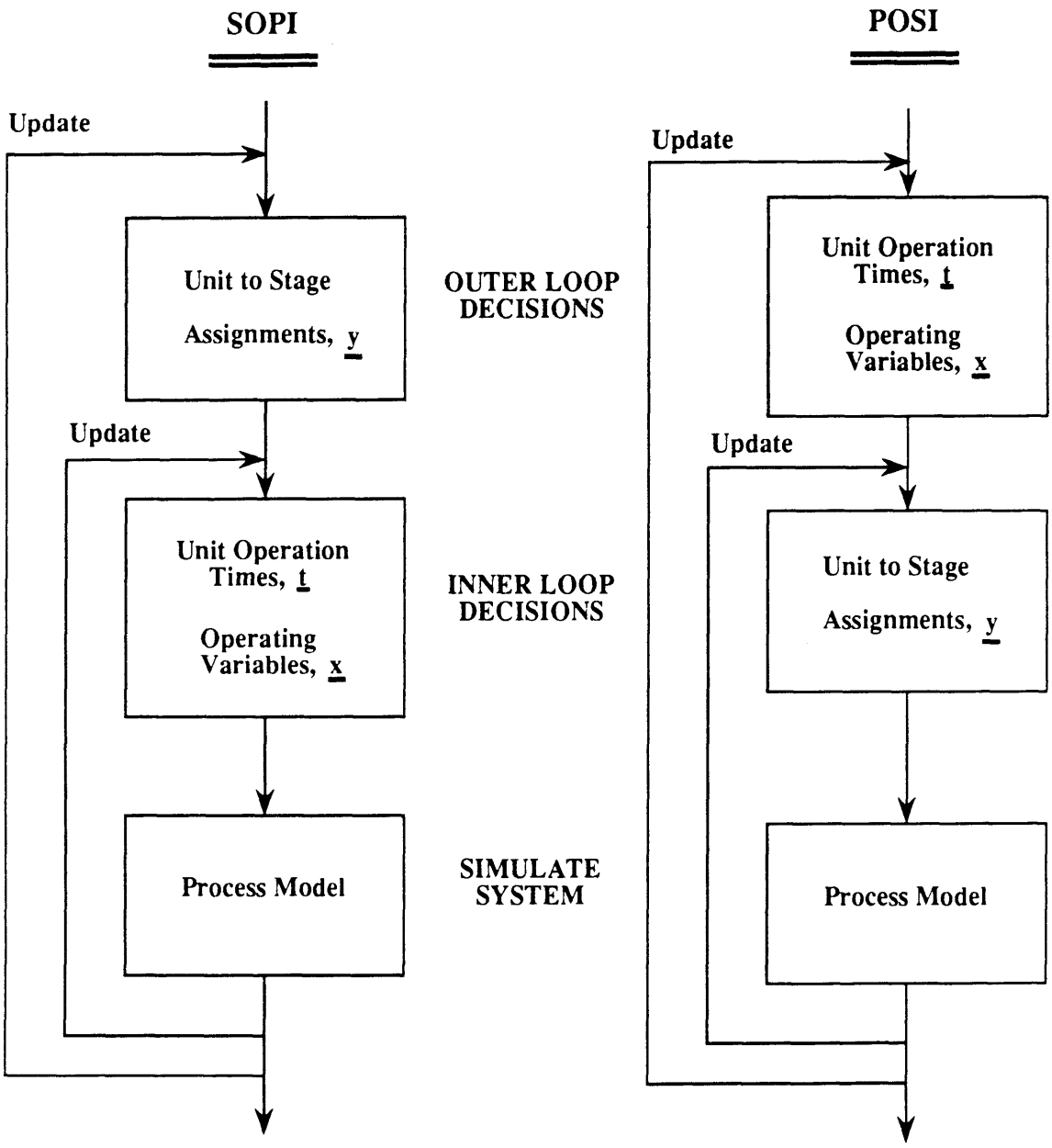


FIGURE 6-6. TWO ALTERNATIVE NESTING STRATEGIES

of each structure now involves a performance optimization rather than a simple cost calculation.

One definite advantage for the SOPI method for this thesis is that the solution procedures used in previous chapters for the individual subproblems can be employed here without the need for significant alterations. Extending the SOPI approach to multiple products is also easily accomplished. By specifying the structure in the outer loop, the performance subproblem is carried out in the inner loop using either the multiproduct NLP formulation or the short-cut method based on the solution of single product performance subproblems. In addition, the SOPI nesting arrangement lends itself nicely to parallel computing opportunities. Performance subproblems for multiple structures or multiple products could be carried out simultaneously by parallel processors.

The issue of computational load presents a potential problem for the SOPI method. Since one performance subproblem has a much heavier computational load than one structure evaluation, putting the performance subproblem in the inside loop could cause the solution times to be quite large. However, the issues of computational effort and solution quality are not easily resolved without carrying out some test problems.

6.3.3 POSI Nesting Arrangement

The POSI method involves nesting the structure subproblem inside the performance optimization. This method is not as aesthetically pleasing as the SOPI method because a combinatorial problem is now embedded inside a continuous optimization problem. Putting the discrete decisions inside the continuous ones causes a few serious difficulties.

These problems are illustrated with the help of Figure 6-7, which shows a schematic of the objective functions for four possible structures plotted as a function of a single performance variable. With the structure subproblem in the inner loop, each evaluation of the objective function in the outer loop gives the value associated with the lowest cost structure at the given value of the performance variable. This result assumes that the structure subproblem always gets the best structure. In this case, the composite objective function being searched by the performance optimization method in the outer loop corresponds to the curve shown in Figure 6-8.

The overall objective function now has discontinuities in its derivatives that occur when switching from one lowest cost structure to another as changes are made in the values for the performance decision variable. In the general case with N performance decision

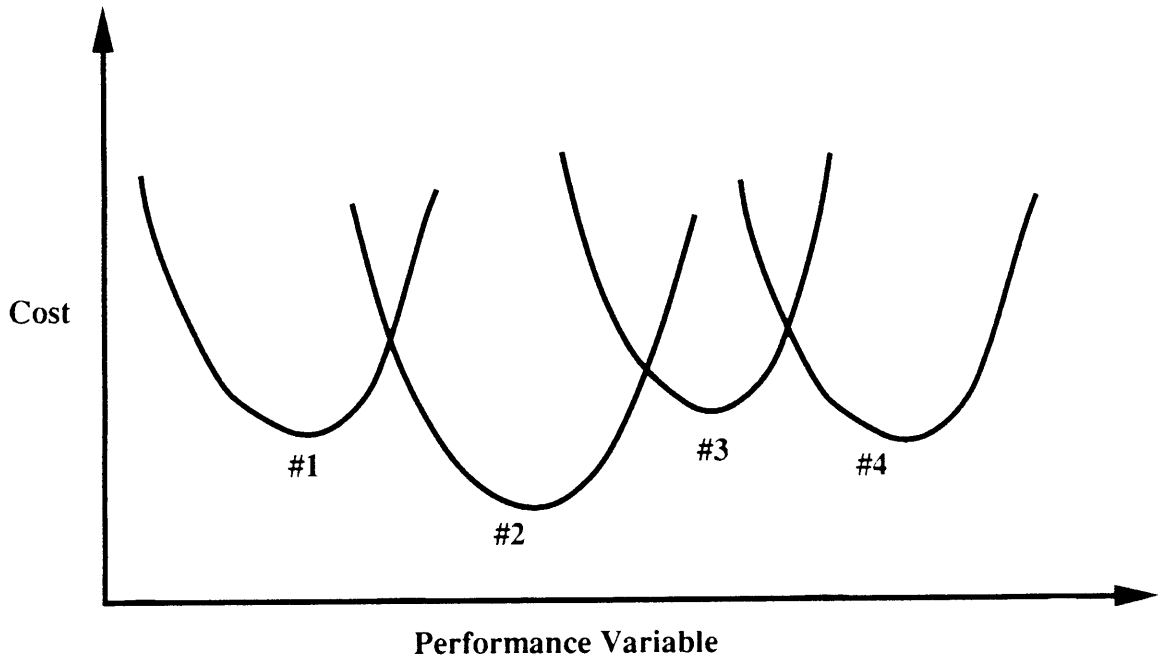


FIGURE 6-7. SCHEMATIC OF COST CURVES FOR 4 DIFFERENT STRUCTURES

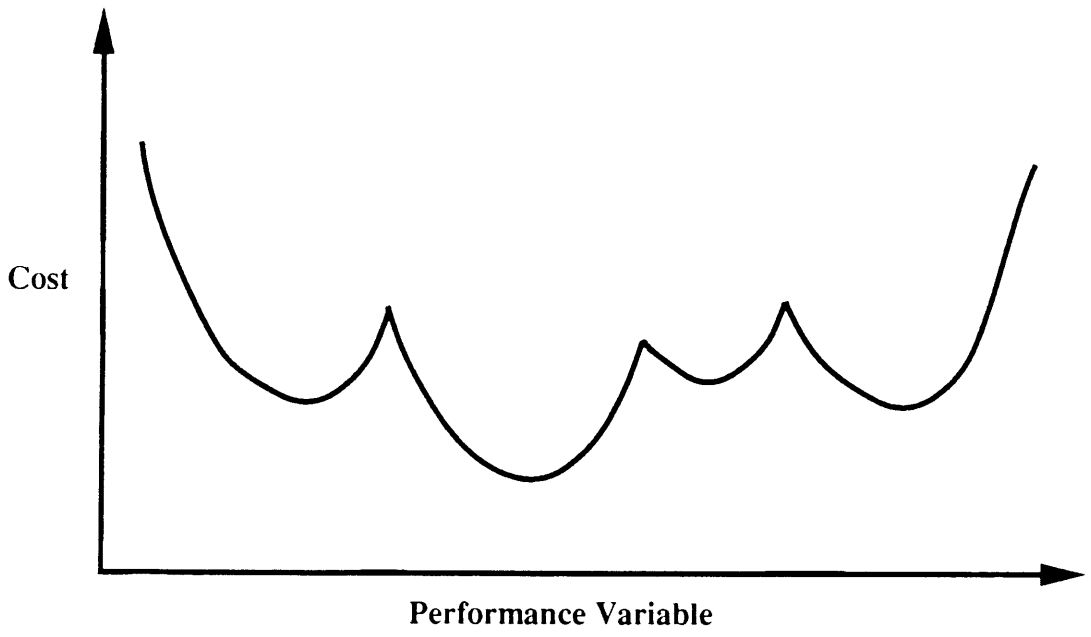


FIGURE 6-8. SCHEMATIC OF COMPOSITE COST CURVE FOR 4 STRUCTURES

variables, a bumpy, N-dimensional surface would represent the overall objective function. Gradient-based optimization methods generally assume unimodal objective functions with smooth first and second derivatives. Since these conditions do not hold for this problem, a new choice must be made for the algorithm to use when optimizing the performance.

If approximate methods, rather than "exact" MINLP or branch and bound methods, are used for the structure subproblem in order to take advantage of relatively short computation time requirements, even more discontinuities are likely. If the approximate method does not get the best structure, there could be large jump discontinuities in the objective function itself in addition to the problems with the derivatives discussed above. These additional discontinuities would cause even greater problems for gradient-based methods trying to optimize the performance decisions in the outer loop.

Another potential problem with the POSI nesting arrangement is that it does not extend to multiple products as easily as the SOPI method. Because the number of performance decisions increases approximately linearly with the number of products, the dimensionality of the search space in the outer loop grows rapidly as the number of products increases. It is also not easy to decompose the multiproduct problem into a series of single product ones with the performance optimization in the outer loop.

Despite all of these difficulties, the POSI arrangement warrants consideration because of the potential for reduced computational effort. If process performance is relatively independent of unit size, then reduced models could be constructed to determine the average rate and total cost for a given set of processing conditions. These reduced models could be used to evaluate the costs for many structures without the need to do a full process simulation. If the simulations required to evaluate process performance are much more time-consuming than the cost calculations required to evaluate a proposed structure, a significant reduction in overall computational effort could be realized by using a calculational procedure like the one shown in Figure 6-9. By putting the process simulations required to determine performance in the outer loop of the optimization, there is the potential to obtain large reductions in computation time if a reasonably efficient method for optimizing the performance in the outer loop can be implemented. Hopefully, the POSI method performs only one set of simulations to search over the possible ranges of decision variables and optimize performance, picking out the best structure as it goes rather than doing a performance optimization for a large number of structures as with the SOPI nesting arrangement.

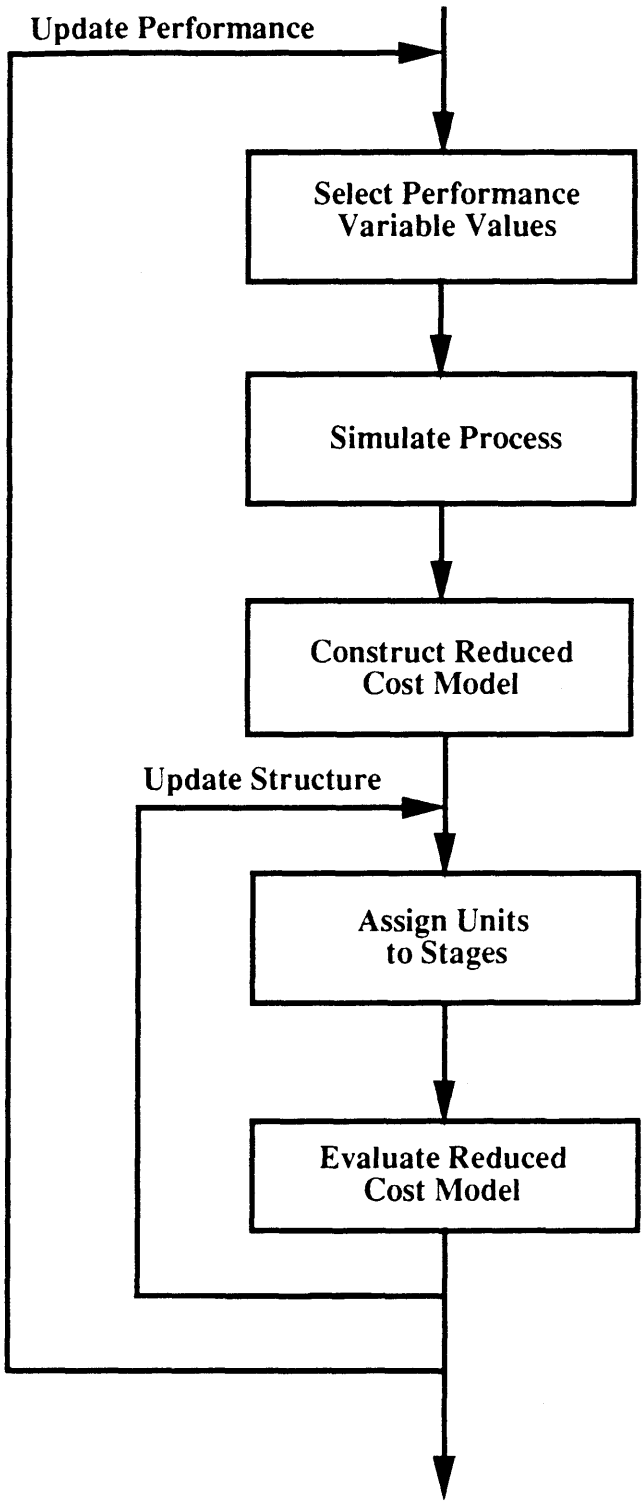


FIGURE 6-9. POSI NESTING ARRANGEMENT WITH USE OF REDUCED COST MODELS

6.4 Description of Test Problems

6.4.1 Introduction

Two simple single product example problems are used to compare these two nesting strategies. The main objective is to determine the relative merits of the two embedding arrangements on problems small enough that all performance subproblems can be enumerated to determine a global optimum. The two main criteria for appraising the performance of the two alternative schemes are: (1) final closeness to the overall optimum, and (2) the computational effort in terms of the number of process simulations. These two example problems should help to identify the key issues involved in developing a successful solution strategy for the combined performance, structure optimization problem. General descriptions of the two test problems are provided in this section.

6.4.2 One Stage Problem

The one stage problem consists of a batch reaction in which reactants A and B form product C and unwanted byproduct D. The reaction is first order in both A and B and takes place at constant temperature. The performance optimization variables are the reaction time, the reaction temperature, and the feed ratio of B to A. Constraints for the minimum concentration of product C at the end of the reaction, the maximum reaction temperature, and the maximum campaign time for producing a given quantity of product are specified. The cost model includes terms for raw materials, utilities, reactor clean-outs, and equipment usage.

An inventory of five reactors is available for use. Each of these units has a different capacity and equipment usage charge. A total of only 31 combinations of units are possible for this one stage problem. This small number of structures makes enumeration of all possible structures quite easy. Appendix A.4 provides more details on the model equations and parameter values used for this example problem.

6.4.3 Two Stage Problem

The two stage test problem for this chapter is an extension of the one stage problem described in Section 6.4.2. Essentially, a separation step has been added to purify product C. A perfect splits column model is used to keep computational requirements low. The magnitudes of the relative volatilities mean that the first material removed by the column is

unwanted byproduct D. Product C is removed second. Unconverted reactants A and B are then sent for further processing and possible recycle. Because the perfect splits assumption is used, the distillation time depends directly on the conversion in the reactor. As in the one stage problem, the three performance optimization variables are the reaction time, reaction temperature, and the feed ratio of B to A.

The inventory of equipment units available for the two stage problem consists of the same five reactors as in the one stage case plus four distillation columns. With 15 possible combinations of column assignments, there are a total of 465 (31 x 15) possible structures for this two stage process. This number is over an order of magnitude larger than the one stage case, so the effects of a larger structure subproblem can be observed. However, the number of structures is still small enough that enumeration remains feasible. Additional details on the process and cost models and the parameter values used for this example problem are given in Appendix A.5.

6.5 Solution Procedures for Test Problems

In order to actually solve the two example problems, solution procedures for the various subproblems must be identified. Although this task is fairly straightforward for the SOPI case, some additional effort is required for the performance optimization for the POSI arrangement. A description of the solution procedures to be used on the two test problems is provided in this section.

6.5.1 Algorithm for SOPI Arrangement

The algorithms used in the SOPI nesting strategy are essentially the ones used for the individual subproblems studied in Chapters 4 and 5. For the performance optimization in the inner loop, the SQP optimization procedure (see Section 4.4) is used to solve the NLP optimization problem. Two methods are used for the structure optimization. First, complete enumeration of the structures is used because it guarantees that the global optimum solution is found. Enumeration also provides a basis of comparison in terms of computational load. The PR/LS procedure (see Section 5.6) is second structure improvement method employed. The PR method is used to get the initial structure because the LB method requires some estimate of appropriate processing conditions to get average rates. Initially, the LS procedure attempts to find a feasible solution by minimizing campaign time. After obtaining a feasible structure, moves or switches are accepted whenever the total cost is reduced.

A minor adjustment is made to the LS procedure to reduce the computational effort. The optimized cost of each structure evaluated in the outer loop is saved. Since a "time-consuming" performance subproblem must be carried out on each outer loop iteration, the list of previously evaluated structures is checked before optimizing the performance of the current structure. Table look-ups eliminate the extra simulations associated with optimizing any structure more than once.

6.5.2 Algorithm for POSI Arrangement

The structure subproblem for the POSI nesting arrangement is solved with the approximate PR/LS method, which obtains near-optimal structures for given performance conditions. Because the processing conditions are known at the outset of any structure subproblem for this embedding scheme, the LB initial structure heuristic could easily be used in place of the PR method. However, the PR method is used to be consistent with the SOPI method for these test problems. Complete enumeration of structures is also used for the one stage problem because the number of structures is so small.

Identifying an appropriate algorithm for the performance optimization in the POSI arrangement is not accomplished so easily. Because of the discontinuities in the objective function, the SQP method used for the performance subproblem is unacceptable. The lack of reliable gradient information eliminates most optimization techniques for continuous optimization problems. The performance optimization approach used for these two example problems is based on insights obtained from the two stage problem considered in Section 6.2.

As discussed previously, the composite total cost surface is expected to be a bumpy, non-unimodal surface. As shown in the schematics in Figures 6-7 and 6-8, the composite objective function is made of sections from many different structures over the range of potential performance variable values. Depending on the location selected in the performance variable space, one particular structure dominates all the rest. Within a small domain of the performance variable space, gradient based optimization techniques could be used to optimize the objective function of the dominant structure. This performance optimization would yield a local optimum solution, unless the dominant structure in this particular region of the performance variable space also happened to be the best overall structure.

The question then becomes how to determine the proper location in the performance variable space in which to lock onto a dominant structure. Although the composite objective function has the bumpy character noted above on the small to medium scale, it seems likely that it might have some large scale behavior that could be exploited. Figure 6-4 shows that there is indeed a large scale trend to the the optimal total cost for the 441 possible structures for the two stage problem in Section 6.2. The composite total cost appears to be some type of roughly quadratic function of optimal reaction time. There is clearly a region in the reaction time space which is best. In other words, if the small scale jaggedness of the composite cost function could be smoothed out, gradient information on the large scale could be used to find the general region of the performance variable space in which most of the best structures have their optimum processing conditions. An implicit assumption in this approach is that the "smoothed" cost function is unimodal on the large scale.

A pattern search approach is used for these two example problems. Although not implemented here, more sophisticated methods, such as surface response techniques, could be tried in an effort to obtain better convergence results. This pattern search method represents a first pass attempt in order to obtain some feel for the potential of the POSI nesting arrangement. The pattern search strategy involves three phases: (1) initialization, (2) grid reduction, and (3) termination. These three phases are described below.

Because trends in the composite objective function are expected only on the large scale, it is desired to start the pattern search with the widest possible range of performance variable values. Because the number of grid points to evaluate increases rapidly with the dimensionality of the search space (i.e., the number of performance variables), it is also desirable to reduce the number of decision variables down to a minimum number by fixing the values of the less significant ones. The key performance variables and the ranges in values to be considered are identified by doing a series of initial performance subproblems using the SQP optimization method.

Structures with the smallest unit(s) and the largest possible number of parallel units are considered in order to estimate the optimal processing conditions for all possible extremes in bottleneck rate and bottleneck location. For the two stage problem, four performance subproblems are required. These problems had the following structures: (1) smallest reactor with smallest column, (2) smallest reactor with all columns in parallel, (3) all reactors in parallel with the smallest column, and (4) all reactors in parallel and all

columns in parallel. The minimum and maximum optimal values for all performance variables from these performance subproblems determine the initial ranges for the pattern search. Since the optimal feed ratio of B to A does not vary significantly for either problem, this performance variable is fixed at a nominal value. Only the reaction time and reaction temperature are optimized using the pattern search.

A two variable pattern search is carried out to optimize the reaction time and reaction temperature for the two test problems. An initial grid of points is established between the bounds obtained from the initialization procedure. Each point in the grid is evaluated by doing a process simulation and then solving the structure subproblem. The "best" grid point is identified, and a new, smaller grid is created around this point. Although the various grid points could be used to develop an approximation of the "smoothed" composite surface, this variation is left for future studies. Series of grids of decreasing size are constructed until a specified termination condition is satisfied.

A number of termination conditions are possible. One potential termination procedure involves stopping when no further reductions are obtained in the objective function, or when the size of the reductions becomes negligible. A second stopping procedure involves kicking out when the ranges of the performance variables in the grid are less than some specified amount. For example, if the minimum and maximum reaction times in the current grid differ by only 0.1 percent, one might decide that further optimization is not warranted. As each performance variable satisfies this termination condition, it would be fixed at the value corresponding to the best solution, and the dimensionality of the grid would be reduced. When all performance variables have been "fixed" in this fashion, the optimization is complete. This second termination procedure is used on the one stage problem.

A third alternative involves recognizing when the grid essentially switches from the large scale to the local scale. Basically, when all the points in the current grid have the same "best" structure, that structure could be considered the dominant structure in the performance variable region of interest. At this point, the overall optimization would be completed by optimizing the performance of this particular structure by doing a performance subproblem. The assumption is that the grid search identifies the good region of the performance variable space. Then, the best structure in this space is fine-tuned by doing a performance subproblem. Another variation of this approach would be to optimize the performance of the best M structures found during the grid search. M would be selected to trade-off the increased computational effort of doing additional performance

subproblems against the greater likelihood of finding the overall best structure. The dominant structure termination procedure is used for both test problems.

6.6 Results of Test Problems

The results for the two test problems are summarized in this section. The SOPI and POSI nesting arrangements are compared in terms of solution quality (closeness to the global optimum) and computational load (number of process simulations and cost evaluations). The global optimal solution is obtained by solving the performance subproblems for all possible structures. Table 6-2 provides a list of all the various combinations of methods used to solve the test problems.

Essentially all the combinations of methods tried for the two alternative nesting arrangements successfully obtain the globally optimal solutions. The best structures are properly identified for both problems. The optimization results are shown in Table 6-3 for the two test problems. The pattern search that terminates when the ranges of values for the performance variables become small (Pattern Search #1) has slightly different final values for the performance variables for the one stage problem, but the total cost is essentially the same as the optimum value. These results mean that no discrimination between the various methods is possible based on solution accuracy.

Although all the methods get the optimal solutions, they do require differing amounts of computational effort. Table 6-4 shows the number of process simulations and cost evaluations done by each of the methods. A number of important results and trends are observed in this comparison of computational effort.

First, large reductions (50 to 60 percent) in the total number of cost evaluations are obtained by using the POSI embedding strategy rather than the best SOPI method. The computational time savings with the POSI approach would be even greater (75 to 85 percent) for cases when the time requirement for a full process simulation is much longer than the time requirement for a simple cost evaluation. Also, the "trend" in relative computational load appears to favor the POSI approach as problem size increases in terms of the number of possible structures.

A second interesting result is seen by examining where the computational effort occurs for the POSI arrangement. A large fraction of the full process simulations for the POSI methods take place during performance subproblems done for initialization or termination. If the ranges of performance variables could be established without the need

TABLE 6-2.
LIST OF ALGORITHMS USED ON TEST PROBLEMS

METHODS FOR IMPROVING PERFORMANCE

Abbreviation	Name	Description
SQP	SQP Algorithm	Successive quadratic programming method solves NLP formulation for the performance subproblem.
PS-1	Pattern Search - 1	A patern search is carried out as described in Section 6.5.2 Initialization requries the solution of a number of performance subproblems, which are done with SQP method. termination is by negligible change in operating conditions.
PS-2	Pattern Search - 2	This variation is the same as PS-1, except that the "dominant structure" termination procedure is used. The performance subproblem requried for termination is solved using the SQP method.

METHODS FOR IMPROVING STRUCTURE

Abbreviation	Name	Description
ENUM	Enumeration	All possible structures are enumerateed to find the best one.
LS-1	Local Search-1	The approximate local search procedure (PR / LS) is used to find near-optimal structures.
LS-2	Local Search-2	This version of the approximate local search procedure uses table look-ups to avoid repeated solution of performance subproblems.

TABLE 6-3.
RESULTS OF TEST PROBLEMS FOR ALTERNATIVE NESTING ARRANGEMENTS FOR COMBINED PROBLEM

1 STAGE PROBLEM			OPTIMAL RESULTS				
Nesting	Structure Method	Perf. Method	Reaction Time	Temp.	B / A Ratio	Cost	Sturcture
SOPI	ENUM	SQP	40.43	453.8	1.029	9.733	R-45
SOPI	LS-1	SQP	40.43	453.8	1.029	9.733	R-45
POSI	ENUM	PS-1	40.29	454.2	1.028	9.733	R-45
POSI	LS-1	PS-1	40.29	454.2	1.028	9.733	R-45
POSI	ENUM	PS-2	40.43	453.8	1.029	9.733	R-45
POSI	LS-1	PS-2	40.43	453.8	1.029	9.733	R-45
2 STAGE PROBLEM			**OPTIMAL RESULTS**				
Nesting	Structure Method	Perf. Method	Reaction Time	Temp.	B / A Ratio	Cost	Sturcture
SOPI	ENUM	SQP	20.51	480	1.010	9.925	R-2345, C-3
SOPI	LS-1	SQP	20.51	480	1.010	9.925	R-2345, C-3
POSI	LS-1	PS-2	20.51	480	1.010	9.925	R-2345, C-3

TABLE 6-4.
COMPUTATIONAL EFFORT ON TEST PROBLEMS FOR ALTERNATIVE NESTING ARRANGEMENTS

1 STAGE PROBLEM			RESULTS				
Nesting	Structure Method	Perf. Method	Total # Simulations	Total # Costings	Number Structures or Grid Points	# Simulations for Start-up	# Simulations for Termination
SOPI	ENUM	SQP	5220	5220	31	--	--
SOPI	LS-1	SQP	4336	4336	28	--	--
SOPI	LS-1	SQP	2756	2756	17	--	--
POSI	ENUM	PS-1	445	1435	33	412	0
POSI	LS-1	PS-1	445	1099	33	412	0
POSI	ENUM	PS-2	608	1104	16	412	180
POSI	LS-1	PS-2	608	1048	16	412	180

2 STAGE PROBLEM			RESULTS				
Nesting	Structure Method	Perf. Method	Total # Simulations	Total # Costings	Number Structures or Grid Points	# Simulations for Start-up	# Simulations for Termination
SOPI	ENUM	SQP	47345	47345	465	--	--
SOPI	LS-1	SQP	6210	6210	57	--	--
SOPI	LS-2	SQP	4420	4420	41	--	--
POSI	LS-1	PS-2	2236	2236	42	420	120

for performance subproblems, additional computational savings might be obtained by eliminating the need for performance subproblems during initialization. Using a termination procedure other than the dominant structure approach could also save on the number of full process simulations. However, some loss of final accuracy might be suffered as a result. Additional work is required to determine the best initialization and termination procedures.

The use of table look-ups to avoid re-optimizing performance for repeated structures pays large dividends for the SOPI arrangement. Reductions of approximately 30 percent in the number of full process simulations are obtained for both test problems. Finally, as expected, the approximate local search method is vastly superior to enumeration for the structure optimization, especially for the two stage problem which had the larger number of possible structures. Even larger savings would be expected for bigger problems.

6.7 Key Issues for Algorithm Development

The results of the two test problems described above would seem to clearly favor the POSI nesting arrangement. Since both nesting arrangements perform equally well on the two test problems in terms of solution quality, the superiority of the POSI arrangement over the SOPI embedding scheme arises solely from its significantly lower computational requirements. However, a clear-cut recommendation of one strategy over the other is not possible until they have been tested on larger problems. It is uncertain that both arrangements will continue to perform in a similar manner in terms of solution quality on larger problems. The SOPI method has many intrinsically satisfying attributes that the POSI arrangement lacks. These characteristics would appear to favor the SOPI method over POSI arrangement in terms of solution accuracy for larger problems.

The superiority of the POSI method in terms of computational effort hinges in part on the ability of the performance optimization in the outer loop to converge. As problems grow in size, the number of performance variables in the outer loop will increase, causing the dimensionality of the search space to go up as well. The POSI strategy is expected to have greater difficulty efficiently searching the discontinuous objective function surface than the SOPI method, which searches over the discrete variables in the outer loop. Cases with multiple products also favor the SOPI approach, which is easily extended to multiple products.

The crux of the matter comes down to a choice between computational effort and expected solution quality. The SOPI method is favored in terms of expected solution quality, but it is slow. The POSI method is fast, but it is unclear how it will perform on larger problems, particularly in terms of solution accuracy. Before any definitive recommendation of one arrangement over the other can be made, additional work must be done to refine the two approaches and better characterize their performance on a wider range of problems.

A large amount of development work needs to be done in order to evaluate the potential of the POSI nesting strategy. First, an efficient outer loop algorithm must be developed in order to optimize process performance variables. There are a number of issues which must be settled, including the selection of appropriate initialization, search, termination procedures. Then, the performance of the outer loop algorithm must be characterized on a range of test problems to determine the effects of the number of structures, the number of performance variables, the number of products, and the quality of the structure optimization in the inner loop.

For the SOPI nesting strategy, appropriate algorithms for the subproblems are more readily available. The biggest drawback to using the SOPI method for large, complex problems is that a large, complex performance subproblem must be solved on each iteration of the outer loop. As discussed in Section 4.7, these performance subproblems are expected to be extremely time-consuming. Thus, future work for the SOPI arrangement must focus on the development of short-cut models and optimization speed-ups. The many conceptually pleasing aspects of the SOPI arrangement would probably make this approach the nesting strategy of choice if the computational load for the performance subproblems can be made more manageable.

6.8 Chapter Summary

This chapter has considered the combined design problem when both performance and structure issues are included in the optimization analysis. Significant interactions between process performance and process structure have been demonstrated by solving a small example problem. Therefore, optimal design procedures must consider both aspects simultaneously or in some type of coordinated fashion. A decomposition strategy has been proposed, with two alternative nesting strategies considered for the performance and structure subproblems. Two example problems have been solved to compare the

performance of these two embedding schemes. The key issues in developing better algorithms for both nesting arrangements have also been identified. Areas for future work on solution approaches for this combined problem are also discussed.

Chapter 7

CASE STUDY DESIGN PROBLEM

7.1 Introduction

The main focus of this thesis is the inclusion of performance issues in the optimal design of batch processes. Fairly simple example processes have been used in previous chapters to illustrate the various trade-offs introduced by including performance issues. These problems have also been used to demonstrate possible solution approaches. Industrial batch processes are likely to be far more complicated in terms of the process behavior and the number of process operating variables. A more realistic design problem is considered in this chapter to show how the concepts and methods described in previous chapters apply to bigger and more complicated design problems like those encountered in industry.

More specifically, this chapter addresses three questions. First, do the same types of performance and structural trade-offs and interactions that were demonstrated previously for simple example problems exist for "industrial-size" problems? Second, can the use of more detailed models and the application of systematic methods to explore these trade-offs and interactions result in significant improvements in design? Finally, can the solution approaches described earlier can be successfully applied to more realistic, industrially sized problems? If not, what changes might be made to make the solution methods more suitable?

The case study problem to be considered in this chapter has been developed for use in a senior design course at M.I.T. The design problem is based on an actual process under development by a large chemical company. Company representatives have provided basic information about the general nature of a process that was undergoing scale up at that time. However, a number of modifications have been made to disguise the product, the process, and the specific company in order to protect proprietary interests. These changes are described in more detail in subsequent sections. The main point to be emphasized here is that no direct comparison is possible with the final process developed by the company

because the case study design problem does not match the actual industrial design problem in most quantitative details. However, the case study problem does have the flavor as well as the size and scope of the industrial problem upon which it has been based.

The case study problem involves the design of a batch process for the production of a mixture of two chemical intermediates (A and D). These intermediates are required in the production of a new specialty product, which has been named Lucretex. This new product has been successfully produced in both the laboratory and the pilot plant. Management now desires to test market Lucretex. In order to produce the required amount of Lucretex, 136,080 kilograms of intermediates A and D are needed.

The design problem involves taking information about the process at the pilot scale, choosing values for all the processing variables and operating times, and selecting the equipment units from an inventory of units available in an existing plant. Several processing constraints must be satisfied, including completion of the campaign within a specified time horizon. The design objective is a final process that satisfies all the given constraints with minimum operating costs.

The chapter is organized as follows. First, a general description of the process is provided. The six tasks in the process and the models that have been used to simulate them are described. Results of a base case design are given to provide a reference for later optimization efforts. Next, the size of the optimization problem and the impact that this has on the choice of solution approach are considered. Because the solution approach used for this case study problem differs from the methods described in previous chapters, these modifications and the rationale behind them are discussed. The most important process trade-offs and areas for improvement to the base case are summarized before presenting the final solution. The chapter closes with some concluding remarks are made regarding the lessons learned from this case study exercise.

7.2 Process Description

The process to be designed produces a mixture of intermediates A and D. The chemicals involved in the case study process have been modified from those in the industrial process by making molecular substitutions. The various materials are also described by pseudonyms throughout this chapter. Figure 7-1 shows a block diagram of the six tasks involved in the production of A and D, of which half are batch reactions and

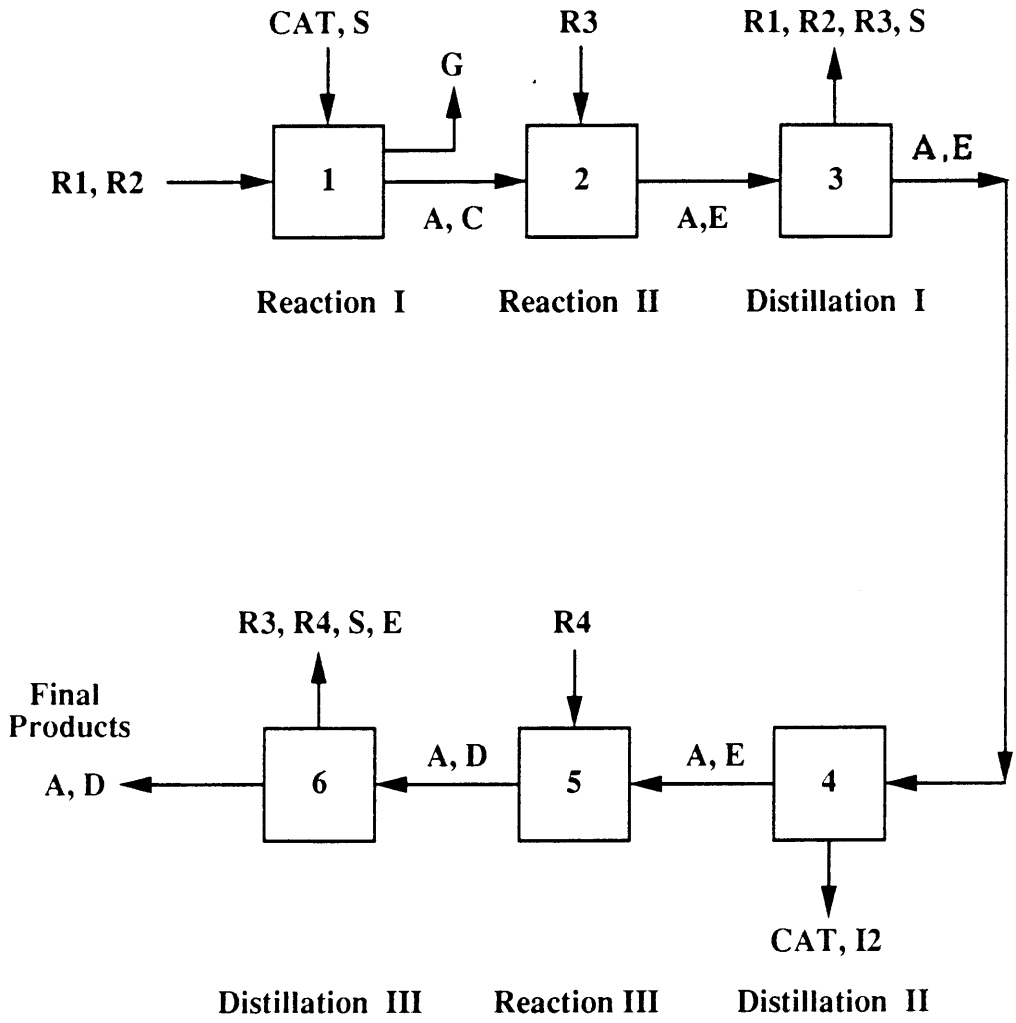


FIGURE 7-1. BLOCK DIAGRAM OF THE SIX TASKS IN THE CASE STUDY DESIGN PROBLEM

half are batch distillations. The six tasks are summarized in Table 7-1 and are described in the paragraphs below.

Reaction I is the first task in the process and the most complicated of the three reactions. Two competing overall reactions occur which convert reactants R1 and R2 into A and C, an intermediate on the way to forming D:

$$2\,R1 + R2 \rightarrow A \qquad (7\text{-}1)$$

$$R1 + R2 \rightarrow C + G \qquad (7\text{-}2)$$

G is an uncondensible byproduct of the second reaction and must be vented. These reactions take place in solvent S in the presence of a catalyst CAT. The reactions are exothermic, and pilot plant runs have been operated at total liquid reflux to facilitate stable operation.

A network of elementary reactions has been postulated for Reaction I in order to develop a kinetic model. This reaction network and the necessary rate constants are assumed because company representatives would not provide detailed kinetic information. The reaction network for Reaction I is shown below:

$$R1 + R2 \rightarrow I1 \qquad (7\text{-}3)$$

$$R1 + I1 \rightarrow A \qquad (7\text{-}4)$$

$$I1 \rightarrow C + G \qquad (7\text{-}5)$$

$$I1 + C \leftrightarrow I2 \qquad (7\text{-}6)$$

$$CAT \rightarrow CAT^* \qquad (7\text{-}7)$$

Compound I1 is a reactive intermediate, and I2 is an unwanted byproduct. The catalyst affects only the reaction described by Equation 7-3. The catalyst activity decays over time as shown by the reaction in Equation 7-7. CAT* represents inactive catalyst.

The set of differential equations describing the kinetics are given in Appendix A.6. The kinetic rate constants are given by Arrhenius type expressions. Values have been assumed for pre-exponential factors and activation energies. The kinetic equations are assumed to hold only when the reactor temperature exceeds 60 °C. A different set of reactions takes place at temperatures below 60 °C. The minimum amount of solvent S required is given as a function of the total amount of R2 that is fed. The reaction volume is also assumed to be independent of conversion. Finally, R2 is assumed to be the limiting reactant, and a conversion specification is imposed on R2.

The second task, Reaction II, involves converting intermediate C into intermediate E by reaction with R3:

TABLE 7-1.
OVERVIEW OF PROCESS TASKS FOR CASE STUDY PROBLEM

Reaction I	Final product A and intermediate C are produced from reactants R1 and R2 in the presence of catalyst CAT and solvent S. Unwanted byproducts G, I1, and I2 are also formed.
Reaction II	Intermediate E is formed from intermediate C through reaction with R3.
Distillation I	Light components (R1, R2, R3, and S) are removed overhead for possible recycle. Small amounts of C and I1 are also removed. The pot contains S, E, A, I2, and CAT.
Distillation II	S, E and A are collected overhead and sent on for further processing leaving unwanted heavy components I2 and CAT in the pot. The heavies are then sent to waste treatment.
Reaction III	Product D is formed from intermediate E through reaction with R4. The reaction regenerates R3 as a byproduct.
Distillation III	Final purification of products A and D is achieved by collecting R3, R4, S, and unreacted E overhead. These materials are then sent to waste treatment.

$$C + R3 \rightarrow E \qquad (7\text{-}8)$$

Reaction II takes place at a constant temperature of 75 °C. The rate expression is given in Appendix A.6. Again, changes in volume with reaction are assumed to be negligible. The conversion of C must exceed 98 percent.

After Reaction II, two separations are required. Distillation I removes light materials from the product containing mixture, which remains in the pot. These low boiling materials, R1, R2, R3, C, I1, and some S, are collected for purification and possible recycle. Distillation II removes heavy components I2 and CAT. A mixture of S, E, and A is collected in the overheads and sent for further processing. The unwanted heavies left in the pot are sent to waste treatment. A maximum temperature constraint of 140 °C is imposed on both Distillations I and II in order to prevent the catalyst from initiating unwanted side reactions that rapidly degrade product A.

Reaction III converts intermediate E to final product D by reaction with R4.

$$R4 + 2\,E \rightarrow D + 2\,R3 \qquad (7\text{-}9)$$

The reaction regenerates R3. Reaction III is also an isothermal reaction, but the temperature is a design variable which has an upper limit of 95 °C. A large excess of R4 is required in order to get proper conversion of E to D. The kinetic rate expression given in Appendix A.6 is only valid when the molar ratio of R4 to E exceeds 25.

The last task is the final purification of the products. Distillation III removes R3, R4, and S overhead, leaving a mixture of A and D in the bottoms. The overheads are sent to waste treatment. A purity specification of 99 percent A and D by mass is imposed on the bottoms product. There is no specification on the ratio of A to D in the final product stream.

The process to produce A and D is to be carried out in an existing facility. A set of equipment units has been set aside for use by this process, although not all the units need be used. There are three types of equipment units, including batch reactors, batch distillation columns, and combination units. Combination units are vessels with a column mounted overhead such that both reaction and distillation tasks can be carried out in the units. Thus, all six tasks could conceivably be carried out in a single unit. The inventory of available units is shown in Table 7-2. Each unit has a specified equipment usage charge. All columns have nine stages (including the still as a stage), with constant vapor boil-up rates as shown. An unlimited supply of intermediate storage units is assumed to be available, each with a volume of 250 gallons.

TABLE 7-2.
INVENTORY OF AVAILABLE EQUIPMENT UNITS

Unit Type (Number Available)	Unit Volume (gallons)	Vapor Rate (kmol/hr)	Usage Charge ($/hr)
REACTORS (5)	500	-	50
	500	-	50
	750	-	70
	750	-	70
	1000	-	88
DISTILLATION COLUMNS (4)*	750	15	90
	750	20	98
	1000	20	110
	1250	15	125
COMBINATION UNITS (1)**	2000	10	175
INTERMEDIATE STORAGE	An unlimited supply of 250 gallon storage units are available. Each storage unit has an equipment usage charge of $5 per hour.		

* All four available distillation columns have 8 trays.

** A combination unit is a vessel with an 8 tray distillation column mounted overhead.

7.3 Process Models

In order to simulate the process, models of the six tasks are required. These models take the form of computer programs to describe the operating behavior of the various tasks with respect to time and changes in input conditions and operating variables. These models also require parameter values, such as rate constants and physical property data. This section describes the models used to simulate the tasks in the process and the sources used to obtain values for required parameters.

Computer programs written in FORTRAN simulate the behavior of the three reaction tasks. Reactions II and III are quite straightforward because analytic expressions for final concentrations of reactants can be developed from the kinetic rate expressions. The constant reaction volume assumption makes the development of analytic expressions possible. Including volume changes would require the kinetic rate expressions to be integrated numerically.

Reaction I is quite complicated because the reaction is not isothermal and reactants can be added over time. The differential equations for the reaction kinetics and reaction temperature are numerically integrated using a variable step size version of the explicit Euler integration method. However, the presence of the reflux condenser requires that some vapor-liquid equilibrium calculations also be performed. A bubble point calculation using Raoult's Law and empirical expressions for vapor pressures as a function of temperature is performed to determine the appropriate temperature update. Multiple operating steps are possible, with different feed rates and jacket heating or cooling duties during each operating step.

The overhead reflux condenser is assumed to provide the exact duty required to condense all of the vapor generated (excluding the uncondensible G) without any subcooling. This assumption is somewhat unrealistic because it generates an extremely difficult dynamic control problem. Also, if the condenser does not condense all of the vapor, material might escape to the atmosphere, posing a serious environmental and health hazard. Therefore, safety considerations would dictate that some subcooling always occur. However, this simplifying assumption has been made to ease the computational burden of simulating Reaction I. A maximum limit on condenser duty is imposed.

Parameter values, such as reaction rate constants and heats of reaction, are not readily available for the reactions in this process. Therefore, these values have been

assumed. In practice, these physical quantities could be estimated from laboratory kinetics experiments. A constant value for the heat of vaporization for the reactor contents (independent of composition) has also been assumed. Again, the use of a simplifying assumption eases the computational burden of simulating Reaction I.

The three distillations were simulated with BATCHFRAC, a commercial batch distillation simulation program available from Aspen Technology (1987). This package rigorously simulates the behavior of batch distillation columns by solving the unsteady state material and energy balances. The stiff system of equations is solved using the "inside-out" algorithm developed by Boston et al. (1981). This general purpose package can handle narrow and wide boiling mixtures as well as highly non-ideal systems.

Distillations I and II use "rigorous" thermodynamic models. The vapor phase is modeled with the Redlich-Kwong-Soave equation of state, while the liquid phase is described by the UNIQUAC liquid solution model. For Distillation III, ideal models are used for both phases in order to reduce the computational load required. Pressure drops in the columns are assumed to be negligible, and liquid hold-ups are assumed to be constant at specified values. A 0.5 hour period of total reflux is required to bring each column to "steady state" at start-up and whenever any major changes in reflux ratio are made.

A large number of physical properties are required for the simulation of the reactions and the distillations. Since the compounds being considered are unusual (especially after the molecular substitutions made for disguise purposes), physical property data are not available for most of the materials of interest. Therefore, the physical properties have been estimated using the stand-alone physical property estimation package available within ASPEN PLUS (Aspen Technology, 1988). This package uses group contribution methods to estimate physical properties based on the molecular structure of the components. For this case study project, the estimated physical properties are assumed to be correct. In practice, these estimates could be checked experimentally. Physical property values are shown in Appendix A.6.

Once the process is simulated, the objective function (cost) must be determined. Few specifics are available for developing a cost model that accurately depicts the costs of the actual industrial process. For this reason, empirical cost expressions have been developed. The objective function includes cost terms for raw materials, waste treatment, utilities, and equipment usage. Direct labor could be assumed to be incorporated into the

equipment usage charges. All values for cost parameters have been assumed. The detailed cost expressions and the parameters values used are summarized in Appendix A.6.

7.4 Base Case Design

Before considering process optimization issues, a base case design is developed to provide a point of reference for later designs. Values for process operating variables and operating times have generally been selected in one of two ways for the base case design. Most operating conditions have been set to correspond to reasonable values based on the information available from the pilot plant stage or previous process development work. For units with processing constraints, operating conditions are chosen in order to guarantee that the specification is be satisfied. A five stage process is assumed, with the only merging of tasks occurring at stage 3, where Distillations I and II are performed in the same column. Units have been assigned somewhat randomly to the stages for the base case. The detailed operating conditions and unit assignments are shown in Tables 7-3 and 7-4, respectively.

The process has been analyzed for both the UIS and NIS cases. For the NIS case, Reaction I limits the batch size at 769 kg of equivalent product, and Distillation III limits the cycle time at 30.8 hours. Thus, the NIS average rate is 24.95 kg per hour. In the UIS case, all the units are operated full. However, the distillation times are proportional to the batch size processed by the columns. Adding storage between all stages increases the average rate only slightly to 25.75 kg per hour. The average rate directly affects the total campaign time and thus the total equipment cost.

The key process performance results for the two cases are shown in Table 7-5. As will be discussed later, the selectivity of Reaction I is crucial. The base case produces roughly an equal amount of A and C on a molar basis. The distillation column operating policies result in the recovery of almost all potential product, although the product purity could be improved for Distillations I and II.

The costs of the two alternatives are also shown in Table 7-5. The NIS case is slightly less expensive because the decreased equipment costs caused by shortening the campaign time in the UIS case are more than offset by the cost of the storage. Utilities costs (40.8 %) account for the largest part of the total, followed by the equipment usage costs (33.3 %). Neither of these two alternatives satisfies the horizon constraint of 150 days. Thus, neither design is feasible.

TABLE 7-3.
OPERATING CONDITIONS FOR BASE CASE DESIGN

TASK	DECISION	VALUE
REACTION I	R1 / R2 Ratio	2 mol R1 / mol R2
	S / R2 Ratio	0.16 liters S / mol R2
	CAT / R2 Ratio	1.27 ml CAT / mol R2
	Feed Temperature	85 °C
	Addition Strategy	Charge: 20% R1, R2; 100% S, CAT
		0 < t < 8 hr: Remaining 80% R1, R2
	Reaction Time	12 hr
	Jacket Duty	0 kcal/hr
	Maximum Reflux Condenser Duty	200 kcal/ hr / gal reaction material
REACTION II	R3 / C Ratio	2.12 mol R3 / mol C
	Reaction Time	3.24 hr (98% conversion of C)
	Temperature	75 °C
DISTILLATION I	Reflux Ratio	10.
	Pressure	760 mm Hg
	Cut Location	Pot Temperature reaches 135 °C
DISTILLATION II	Reflux Ratio	10.
	Pressure	5. mm Hg
	Cut Location	Pot Temperature reaches 135 °C
REACTION III	R4 / E Ratio	30 mol R4 / mol E
	Temperature	90 °C
	Reaction Time	3.25 hr (85% conversion of E)
DISTILLATION III	Reflux Ratio	8.
	Pressure	760. mm Hg
	Cut Location	Purity of A & D in pot of 99% (mass)

TABLE 7-4.
PROCESS STRUCTURE FOR BASE CASE DESIGN

STAGE NUMBER	TASKS ASSIGNED	UNITS ASSIGNED
1	Reaction I	500 gallon reactor
2	Reaction II	750 gallon reactor
3	Distillation I Distillation II	750 gallon still 15 kmol/hr vapor rate
4	Reaction III	500 gallon
5	Distillation III	1250 gallon still 15 kmol/hr vapor rate

UNITS LEFT IN INVENTORY:	Reactors (2):	750 gal; 1000 gal
	Columns (2):	750 gal (20 kmol/hr); 1000 gal (20 kmol/hr)
	Combination (1):	2000 gal (10 kmol/hr)

TABLE 7-5.
RESULTS FOR BASE CASE DESIGN

PERFORMANCE MEASURES	NIS CASE	UIS CASE
Reaction I:		
Selectivity (mol A / mol C)	1.083	1.083
% Conversion R2 to A or C	98.27 %	98.27 %
Distillation I:		
% Recovery of A and E (by mol)	93.24 %	93.24 %
Mole Fraction (A + E) in product stream	0.7675	0.7675
Distillation II:		
% Recovery of A and E (by mol)	98.91 %	98.91 %
Mole Fraction (A + E) in product stream	0.7710	0.7710
Overall Molar Yield: (A + 2D) / R2	83.95 %	83.95 %
Bottleneck Average Production Rate	24.95 kg/hr	25.75 kg/hr
ECONOMIC MEASURES	**NIS CASE**	**UIS CASE**
Raw Materials Costs	$ 1363.7 K	$ 1363.7 K
Waste Treatment Costs	$ 268.5 K	$ 268.5 K
Utilities Costs	$ 2580.2 K	$ 2553.6 K
Equipment Usage Costs	$ 2100.1 K	$ 2033.4 K
Intermediate Storage Costs	$ 0.0 K	$ 396.1 K
TOTAL COSTS	$ 6312.5 K	$ 6615.3 K
CAMPAIGN TIME	227.3 days	220.1 days

One possible way to generate a feasible solution would be to re-assign the available equipment units to the various stages in an attempt to increase the bottleneck rate. The approximate method based on local search for the structure subproblem is applied to the base case process with the performance fixed. The structure improvement method obtains a significantly improved solution compared to the original NIS base case structure.

A comparison of the two solutions is shown in Table 7-6. Assigning a larger reactor to Reaction I increases the average rate of that stage. The column with the highest vapor rate is assigned to Distillations I and II, and three columns in parallel are assigned to Distillation III, the limiting rate stage in the original base case. These changes reduce the total cost 16.7 percent because the total campaign time is decreased 70.6 percent. The new campaign time of 66.8 days easily satisfies the horizon constraint of 150 days.

7.5 Scope for Optimization

The scope for optimization for this design problem is quite large. A large number of decisions must be made, including the choices of values for operating variables and processing times, the assignment of tasks to stages, the assignment of equipment units to stages, and the placement of intermediate storage. These decisions cover four of the five areas described in Section 2.2.1. Taking all of these factors into account presents the process designer with a wide range of options to consider. Designing a minimum operating cost process which produces the specified amount of A and D on the available equipment within the given horizon is therefore quite difficult. This section describes some of the features which make the scope for optimization so large.

There are a total of 23 operating variables and processing times which must be determined in order to fix the process performance. These decision variables have been listed in Table 7-3. Some of these decisions are actually "profiles" in the sense that the operating variable could be time dependent, e.g., reflux ratio during a cut. The number of variables given above assumes that only one operating period is used for Reaction I, and that the minimum number of operating steps are used in the columns. Using more sophisticated operating strategies would naturally increase the size of the problem.

Not all of the variables listed in Table 7-3 are independent. A total of 16 performance constraints must be satisfied. These constraints include conversion specifications on Reactions I and II, a minimum solvent requirement in Reaction I, temperature constraints (either bounds or specifications) for all three reactions and two of

TABLE 7-6.
COMPARISON OF IMPROVED STRUCTURE WITH ORIGINAL STRUCTURE FOR BASE CASE CONDITIONS

STAGE NUMBER AND TASKS ASSIGNED		IMPROVED PROCESS STRUCTURE (UIS)	ORIGINAL PROCESS STRUCTURE (NIS)
1	Reaction I	1000 gallon reactor	500 gallon reactor
2	Reaction II	500 gallon reactor	750 gallon reactor
3	Distillations I, II	1000 gal still, 20 kmol/hr	750 gal still, 15 kmol/hr
4	Reaction III	500 gallon reactor	500 gallon
5	Distillation III	750 gal still, 15 kmol/hr 750 gal still, 20 kmol/hr 1250 gal still, 15 kmol/hr	1250 gal still, 15 kmol/hr

ECONOMIC RESULTS	IMPROVED	ORIGINAL
Raw Materials Costs	$ 1363.7 K	$ 1363.7 K
Waste Treatment Costs	$ 268.5 K	$ 268.5 K
Utilities Costs	$ 2474.6 K	$ 2580.2 K
Equipment Usage Costs	$ 980.0 K	$ 2100.1 K
Intermediate Storage Costs	$ 168.4 K	$ 0.0 K
TOTAL COSTS	$ 5255.3 K	$ 6312.5 K
CAMPAIGN TIME	66.8 days	227.3 days

the distillations, and a purity specification for Distillation III. Most of these constraints are inequality or bound constraints as opposed to equality constraints. Table 7-7 lists all processing constraints for the design problem.

There are also many possible structures for this process. In addition, there are a number of possible assignments of tasks to stages. Determining the total number of structures in this case is complicated by the fact that the combination unit can be used for either a reaction or a distillation task. Thus, the assumption of a one to one matching of task to unit types, which was used to derive the expressions for the total number of structures in Section 5.4, does not hold in this case. The presence of identical reactors also makes it more difficult to determine the number of unique structures. Table 7-8 shows all feasible task to stage assignments and the total number of possible structures for each.

Intermediate storage greatly increases the number of alternatives. If the intermediate storage policy between any pair of stages is either NIS or UIS, then there are $2^{(n_s-1)}$ possible mixed storage alternatives for each assignment of tasks to n_s stages. As shown in Table 7-8, factoring storage into the picture causes the total number of possible structures to grow from 217,879 to 4,606,055. The combinatorial explosion in the problem is clearly obvious.

7.6 Solution Approach

Before trying to apply the solution approaches discussed in the previous three chapters, the computational load that would be required for this particular problem is estimated. Clearly, both the performance and the structure aspects of this large combinatorial optimization problem are quite involved. Since the combined problem will be at least as demanding as the solution of either a single performance subproblem or a single structure subproblem, the computational requirements of these two subproblems are estimated first as a lower bound on the computational requirements for the overall problem.

The computational effort required to apply the mathematical programming approach described in Chapter 4 to optimize the performance of any given structure is considered first. For most of the rather simple example problems described in Chapter 4, the number of function calls made by the SQP algorithm was in the range of 20 to 40. Increasing the number of decision variables, N_d, to more than twenty will definitely increase the number of function calls required by the SQP algorithm. Thus, 20 function calls represents a lower bound.

TABLE 7-7.
SUMMARY OF CONSTRAINTS ON PROCESS

Reaction I	1. All solvent S must be charged at the start of the reaction. 2. The ratio of the amount of S charged to the total number of moles of R2 fed must be greater than 160 ml S per mole R2. 3. At the end of the reaction, the ratio of S to R2 must be greater than 8400 ml S per mole R2 remaining. 4. The temperature of the reaction must always exceed 60 °C. 5. The maximum reflux condenser duty must not exceed 200 kcal per hour per gallon of reactor capacity.
Reaction II	6. Reaction temperature is constant at 75 °C. 7. The conversion of C must exceed 98 percent.
Distillation I	8. The still pot temperature must not exceed 140 °C because the catalyst initiates unwanted reactions above this temperature. 9. The lowest possible operating pressure is 2 mm Hg.
Distillation II	10. The still pot temperature must not exceed 140 °C because the catalyst initiates unwanted reactions above this temperature. 11. The lowest possible operating pressure is 2 mm Hg.
Reaction III	12. The reaction temperature is constant and must not exceed 95 °C. 13. The molar feed ratio of R4 to E must be greater than 25.
Distillation III	14. The lowest possible operating pressure is 2 mm Hg. 15. Final pot contents must be at least 99 percent A and D (mass).
Overall Process	16. The process must produce 136,080 kg of A and D in less than 150 days (24 hour per day operation).

TABLE 7-8.
NUMBER OF POSSIBLE EQUIPMENT STRUCTURES

Stages	Task to Stage Assignment	Number of Possible Unit to Stage Assignments	Number of Possible Assignments with Storage Included
1	r1 r2 d1 d2 r3 d3	1	1
	r1 / r2 d1 d2 r3 d3	17	34
2	r1 r2 / d1 d2 r3 d3	17	34
	r1 r2 d1 d2 r3 / d3	15	30
	r1 / r2 / d1 d2 r3 d3	74	296
	r1 / r2 d1 d2 r3 / d3	255	1020
3	r1 r2 / d1 / d2 r3 d3	255	1020
	r1 r2 / d1 d2 / r3 d3	255	1020
	r1 r2 / d1 d2 r3 / d3	255	1020
	r1 r2 d1 d2 / r3 / d3	255	1020
	r1 / r2 / d1 / d2 r3 d3	1110	8880
	r1 / r2 / d1 d2 / r3 d3	1110	8880
	r1 / r2 / d1 d2 r3 / d3	1110	8880
4	r1 / r2 d1 d2 / r3 / d3	1110	8880
	r1 r2 / d1 / d2 / r3 d3	850	6800
	r1 r2 / d1 / d2 r3 / d3	850	6800
	r1 r2 / d1 d2 / r3 / d3	21,720	173,760
	r1 r2 d1 / d2 / r3 / d3	850	6800
	r1 / r2 / d1 / d2 / r3 d3	3700	59,200
	r1 / r2 / d1 / d2 r3 / d3	3700	59,200
5	r1 / r2 / d1 d2 / r3 / d3	52,320	837,120
	r1 / r2 d1 / d2 / r3 / d3	3700	59,200
	r1 r2 / d1 / d2 / r3 / d3	38,940	623,040
6	r1 / r2 / d1 / d2 / r3 / d3	85,410	2,733,120
TOTAL NUMBER OF STRUCTURES		217,879	4,606,055

The number of process simulations is (N_d+1) times greater than the number of function calls because derivatives are approximated by finite difference. Each additional decision variable generates an additional simulation on each function call. For the case study problem, 20 function calls correspond to 480 simulations. If each simulation requires only 5 minutes of cpu time, then 2400 cpu minutes (40 cpu hours!) would be required for the performance subproblem for one structure.

In fact, one simulation of the three reactions and three distillations requires approximately 40 cpu minutes on the DEC Microvax II used for this case study problem. In this case, a lower bound of 19,200 cpu minutes (13.33 cpu days!) would be required for the performance subproblem for one structure. Since the actual number of function calls is expected to be much higher than twenty, it is clear that using a mathematical programming approach on this computer is impractical for optimizing the performance of this case study problem unless measures are taken to speed up the solution procedure or reduce the scope of the optimization problem.

For the structure subproblem, the situation is much better. Although there are a large number of possible structures, the time required to calculate the total cost for a given structure is on the order of cpu seconds on the DEC Microvax II. A reduced cost model is determined for the given performance conditions, eliminating the need to do repeated full simulations of the process. Using the PR/LS/RS approximate solution procedure developed in Chapter 5, the computational load can be estimated to be on the order of cpu minutes for determining a near-optimal solution to the structure subproblem for given performance conditions. This time does not include the time required for the one process simulation needed to determine parameter values which go into the reduced cost model. There are some additional difficulties caused by the one combination unit and the possibility of mixed storage policies. Task to stage assignments also have to be fixed in advance since the local search method assumes that these assignments are given.

The general conclusion that can be drawn from the above discussion is that some significant speed-ups or modifications would be necessary to the method used for the performance optimization in order to obtain an optimized design in a reasonable amount of time. This discussion also provides further support for the development of methods based on the POSI nesting arrangement, with performance issues considered in the outer loop. For this case study, an evolutionary approach has been used to determine a final design,

which is hopefully near-optimal. However, the complexity of the problem makes it impossible to state with certainty how close any solution is to the optimum.

The evolutionary solution strategy used for this case study problem is shown schematically in Figure 7-2. Because of the high computational requirements for simulating the entire process and the relatively short time requirements for doing structure evaluations, the evolutionary approach developed incorporates the POSI nesting strategy. Performance improvements are made in the outer loop by identifying and exploring key performance trade-offs. The approximate procedure developed in Chapter 5 is used for optimization of unit to stage assignments in the inner loop. Other structural improvements, such as intermediate storage and task to stage assignment, require some manual reformulations of the information needed by the local search program.

The evolutionary strategy assumes that a base case solution is available. If a feasible base case design has not yet been found, initial improvements are made in order to reduce infeasibilities. Once a feasible solution is available, the approach consists of the following steps:

(1) Identify key performance trade-offs and the decision variables that directly affect them. Single unit simulations may be performed to establish the trends in the performance trade-offs for that unit.
(2) Select new values for the performance variables.
(3) Simulate the entire process.
(4) Assign tasks to stages.
(5) Determine a good structure by using the approximate procedure based on local search to solve the structure subproblem.
(6) Locate and size intermediate storage.
(7) Determine the total operating costs.
(8) Repeat steps 4 through 7 for a variety of task to stage assignments until no further improvements are obtained.
(9) Repeat steps 1 through 8 until no further improvements are obtained.

This evolutionary approach relies on the engineering judgment of the designer to make appropriate changes in the values of processing times and operating variables in order to reduce the total cost of the process. The process model is used as an analysis tool in the design process. The key to developing a good design is the identification of the dominant performance trade-offs. The interactions caused by the selection of equipment units and the processing conditions also affect the total cost. The chief benefit of this evolutionary

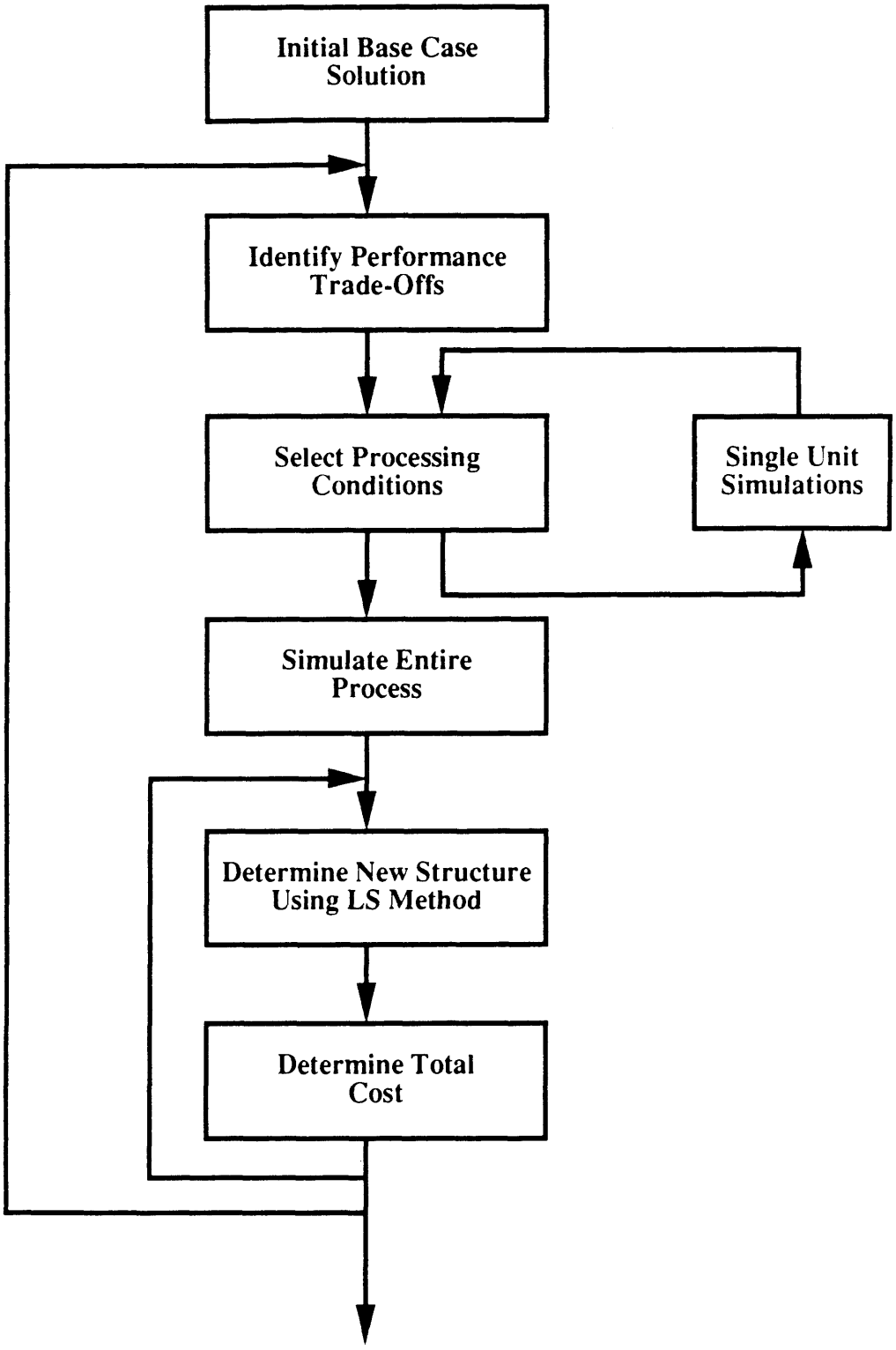


FIGURE 7-2. EVOLUTIONARY SOLUTION STRATEGY FOR CASE STUDY DESIGN PROBLEM

approach is that fewer process simulations are required to get a reasonably good final solution.

7.7 Process Improvements

The starting point for optimization work is the process with the base case operating conditions and the structure obtained by applying the local search method. The evolutionary approach described above is used to make process improvements. Rather than describe in detail all of the incremental design changes in the order that they occur, the most important trade-offs are summarized. An overview of the course of the evolutionary improvements is also provided to give some sense of how the approach works.

Two general areas are targeted for performance improvements. The most important performance decisions involve the operating policies for the three distillations. A second area of attention focuses on the operation of Reaction I. The various trade-offs associated with these two areas are discussed in the following paragraphs.

For the separations in this process, column operating performance is determined by the choices of reflux ratios, locations of the cuts, and operating pressures. The impact of the operating policies of the columns can be seen by examining in detail the results of the base case simulation. Major contributions to the high utilities costs are made by the choices of reflux ratios for the three distillations. Also, the distillations tend to be the low average rate tasks in the base case, causing the campaign time and thus the total equipment costs to increase. There are lesser effects on the waste treatment and raw materials costs.

The major trade-offs with the columns involve the quality of the separations (both in terms of purity and fractional recovery) and the amount of effort required to obtain that level of quality. For example, very sharp splits permit high recovery of products and reactants for recycle and thus lower total costs for raw materials and waste treatment. However, very sharp splits require higher reflux ratios, which increase utility costs and increase the time required to process a batch. The operating pressure can also be adjusted to improve the sharpness of the splits if the relative volatilities of the components vary with pressure. In this particular process, lowering the pressure improves the quality of the separation in Distillations I and II. However, running at vacuum increases operating costs. These trade-offs are explored by varying the reflux ratios and operating pressures for the distillations.

The locations of the cuts for Distillations I and II also affect the performance of the process. For any given amount of effort (specified reflux ratio and operating pressure), the cut location defines a trade-off between purity and fractional recovery. For example, the cuts could be selected to give a high purity of A and E in Distillation II in order to reduce the amount of S which is carried along to Reaction III and Distillation III. However, this high purity would come at the expense of some lost product. Thus, the cut locations involve a performance load trade-off. In addition, the cut locations interact strongly with the reflux ratio and pressure. Therefore, the best cut locations must be determined for every new set of reflux ratios and operating pressures.

A number of single task simulations are carried out to explore the effects of reflux ratio on the sharpness of the splits in the three distillations. A series of BATCHFRAC runs shows that the base case reflux ratios are generally higher than necessary, particularly for Distillations II and III. Initially, large improvements to the base case can be obtained by lowering the reflux ratios for the distillations. The savings in utility and equipment costs greatly outweigh any reductions in the quality of the separations. However, at later stages in the design, the needs for high product recovery and low utility and equipment costs have to be balanced.

A second focus of attention is placed on Reaction I. This step occurs first in the process and has significant effects on downstream tasks. In particular, the final ratio of A to C plays an important role. Increasing the selectivity of product A over intermediate C improves the economics of the process for a number of reasons. First, product A costs less to make than product D (which occurs through formation of C). Assuming complete conversion for all reaction steps in each route, product A is $4.86 per kilogram cheaper to make than D in raw materials and waste treatment costs alone. Second, fewer processing tasks are required to make A compared to coproduct D. Thus, the production of A is preferred over the production of D from a raw materials, waste treatment, and utilities standpoint.

Simulations of Reaction I show that increasing the R1 to R2 feed ratio and decreasing the reaction temperature improves the selectivity of A over C. However, each of these two strategies has costs associated with it. Increasing the R1 to R2 feed ratio increases raw materials and utilities costs because the product yield on R1 decreases and the unconverted R1 must be recovered for recycle. Lowering the reaction temperature increases the time required to reach the conversion specification on R2. Also, since the

reaction is exothermic, cooling must be supplied to prevent the temperature from increasing. The slower reaction rate causes equipment costs to increase if Reaction I becomes the bottleneck unit. Increasing catalyst levels can help increase the reaction rate to a limited extent, but at the expense of higher raw materials and waste treatment costs. Thus, improvements in selectivity must be weighed against potentially higher operating costs.

Many options are available for operating strategies for Reaction I. Reactants can be fed all at once or over time in a series of operating steps. The reaction temperature can also be manipulated by varying the jacket heating or cooling rate over time. However, it is virtually impossible to achieve formation of only A and satisfy the R2 conversion requirement in a reasonable period of time without going to extremely high R1 to R2 ratios. Therefore, appreciable amounts of both A and C are always produced.

Alternative flowsheets could be developed that treat C as an unwanted byproduct. The process could then be reduced to three tasks. Reaction I would be followed by Distillations I and II to produce only product A. Synthesis issues are considered outside the scope of this project and have not been addressed. All proposed designs must carry out the six tasks in the order specified.

Reactions II and III have much less effect on the overall economics than Reaction I and the three distillations. These two reactions are generally operated in order to maximize the product yield obtained in a given amount of time, with the maximum cycle times for these two tasks estimated from the time requirement for Reaction I. The main idea here is to get the most out of these two tasks without making either one of them the bottleneck stage.

Structural trade-offs are considered in the inner loop of the evolutionary approach. The assignment of units to stages has a significant impact on the location and magnitude of the overall process bottleneck production rate, and hence the total equipment costs. This effect was clearly demonstrated with the base case solutions presented in Section 7.4. The general structural trade-offs for this problem are generally the same as those discussed in Sections 2.1.2 and 5.2, and involve balancing the rates of the stages by assigning the appropriate units, merging adjacent tasks, or adding intermediate storage.

The course of the evolutionary approach is summarized below. Because full process simulations are so computationally expensive, very few are carried out. However,

at various points in the course of the analysis, individual units are simulated repeatedly to get a better understanding of the trade-offs affecting the local process performance. Global performance effects are then extrapolated to determine which specific operating scheme appears most promising. This new operating scheme would then be used in the next full process simulation. For each set of performance conditions, efforts are made to determine the best structure. These efforts include the use of the approximate local search method developed in Chapter 5.

Table 7-9 summarizes the major performance changes made on each iteration of the outer loop of the evolutionary approach. The performance results and operating costs for the "best" structure for each of the simulations following the base case simulation are shown in Tables 7-10 and 7-11. Although there is no guarantee that the best design (Simulation #5) is optimal, the evolutionary strategy does succeed in obtaining significant improvements without the need for hundreds of complete process simulations.

The main changes from the base case for Simulation #2 involve adjusting the reflux ratios and cut locations for the three distillations. Operating conditions for Reaction III are also modified to make constraints tight. However, the conditions for Reactions I and II are not altered. Simulation #2 obtains a substantial reduction (27.2 %) in total costs compared to the results for the feasible base case. The main cost reductions occur in the utilities (35.3 %) and equipment (45.1 %) costs. Raw materials costs are cut by only 7.0 percent, and waste treatment costs actually increase 21.5 percent. The increase in waste treatment costs occurs because less S is removed in Distillation I.

The results for Simulation #2 show the beneficial effects of decreasing the reflux ratios. Simulation #3 is carried out after running a number of simulations to determine a better operation strategy for Reaction I. The initial reaction temperature is reduced to 60 °C to favor production of A and reduce reactor heat-up costs. Jacket cooling is employed to slow the temperature rise caused by the exothermic nature of the reactions. However, the reaction temperature still reaches the mixture boiling point quite rapidly. R2 is fed over time while all the R1 is charged initially to favor the formation of A rather than C. Much higher levels of catalyst are used to increase the overall reaction rate. The reflux ratio for Distillation I is increased to improve the sharpness of the cut so that more S would be recovered. Reflux ratios for Distillations II and III are reduced further.

The results for Simulation #3 are somewhat mixed. Although total costs are reduced by 11.9 percent, raw materials (28.1 %) and waste treatment (19.3 %) costs both

TABLE 7-9.
PROCESSING CONDITIONS SIMULATED AT EACH ITERATION OF OUTER LOOP IN EVOLUTIONARY APPROACH

ITERATION	DESCRIPTION OF PROCESSING CONDITIONS
1	Base Case Conditions (details provided in Table 7-3)
2	Major Changes from Simulation #1: 1. Distillation I reflux ratio is reduced to 5 to shorten column operation time. The cut is taken when the pot temperature reaches 115 °C. 2. Distillation II reflux ratio is reduced to 5. 3. Temperature of Reaction III is increased to 95 °C. 4. R4/E ratio in Reaction III is decreased to 25. 5. Conversion of E in Reaction III is increased to 90%. 6. Distillation III reflux ratio is reduced to 4.
3	Major Changes from Simulation #2: 1. Distillation I reflux ratio is increased to 12 to sharpen split. 2. Distillation II reflux ratio is reduced to 2 to shorten the column time. 3. Multiple reflux ratios (1 and 6) are used in Distillation III to shorten the cycle time while retaining a sharp split. 4. Time for Reaction I is shortened to 5 hours with increased catalyst loading and use of a new feeding strategy for R1, R2, and CAT. 5. Reaction I temperature is reduced by charging at 60 °C and using jacket cooling during the reaction. 6. Conversion of C in Reaction II is increased to 98.9% by increasing the reaction time to 6 hours. 7. Conversion of E in Reaction III is increased to 91.9% by increasing reaction time to 6 hours.

TABLE 7-9 continued.

4	Major Changes from Simulation #3: 1. Time for Reaction I is increased to 8 hours, and the catalyst loading is decreased. 2. The feeding strategy and jacket cooling profile are adjusted to increase the selectivity of A over C. 3. The reflux ratios and cut locations for all three distillations are adjusted to reduce operating times and maximize product recoveries. 4. The pressure is reduced to 2 mm Hg for the second half of Distillation I and for all of Distillation II.
5	Change from Simulation #4: 1. Rapid quench is added to the end of Reaction I.
6	Change from Simulation #5: 1. The cut location for Distillation I is adjusted slightly.

TABLE 7-10.
PERFORMANCE RESULTS FOR OUTER LOOP SIMULATIONS

PERFORMANCE	RESULTS FOR SIMULATION NUMBERS: 2	3	4	5	6
REACTION I:					
Selectivity (A / C)	1.083	1.981	2.282	2.282	2.282
Yield of A, C on R2	98.27 %	97.51 %	99.74 %	99.74 %	99.74 %
DISTILLATION I:					
% Recovery A, E	99.43 %	98.24 %	97.43 %	97.43 %	97.91 %
Mol Fraction (A+E)	0.519	0.452	0.980	0.980	0.837
DISTILLATION II:					
% Recovery A, E	98.99 %	92.02 %	99.62 %	99.62 %	99.62 %
Mol Fraction (A+E)	0.519	0.448	0.997	0.997	0.849
Overall % Yield:					
(A+2D)/R2	91.22 %	85.49 %	94.69 %	94.69 %	95.15 %
Bottleneck Average Production Rate (kg A+D / hr)	118.4	146.9	177.7	188.2	183.2

TABLE 7-11.
COST RESULTS FOR OUTER LOOP SIMULATIONS

COSTS	COSTS ($K) FOR SIMULATION NUMBERS: 2	3	4	5	6
Raw Materials	1268.7	1625.4	1215.6	1216.2	1211.3
Waste Treatment	326.2	389.1	121.0	121.1	151.6
Utilities	1623.8	894.1	930.7	799.0	788.9
Equipment Usage	561.0	385.4	297.2	280.6	303.1
Storage Costs	57.5	51.0	53.6	43.4	37.1
TOTAL COST	3837.2	3345.0	2618.1	2460.2	2492.0
% SAVINGS COMPARED TO NIS BASE CASE	39.2 %	47.0 %	58.5 %	61.0 %	60.5 %
CAMPAIGN TIME (days)	47.9	38.6	31.9	30.1	31.0

increased significantly over the costs obtained in Simulation #2. Despite increases in the selectivity to A and the amount of product obtained in each batch of Reaction I, the beneficial effects obtained by increasing the catalyst loading and modifying the addition strategy are more than offset by increased product losses in Distillation II and the increased costs for buying and disposing of the additional catalyst. This unexpected result illustrates how performance load effects can defeat the apparent improvements made at a single stage. Distillation I is again quite ineffective in removing S, resulting in reduced concentrations of A and E sent to Reaction III. The overall cost reduction occurs because lowering reflux ratios reduces utilities costs by 44.1 percent. Also, equipment costs decrease by 27.1 percent because average rates for Reaction I and the distillations increase.

Before running simulation #4, another set of Reaction I simulations is carried out in an attempt to improve the selectivity to A while using lower amounts of catalyst. A new feeding strategy for Reaction I is developed to increase the selectivity of A yet finish in a reasonable period of time. A low temperature operating step promoting the formation of A is followed by a high temperature period to quickly consume the remaining R2.

More complicated operating strategies are also considered for the distillations. In order to increase the sharpness of the splits and reduce product losses, multiple reflux ratios per cut are tried. Because of the interactions between the reflux ratios and the location of the cuts, a number of column simulations are done to determine the best combination of values. Lower pressures in the second half of Distillation I and during Distillation II also increase the sharpness of the splits obtained.

The changes in operating conditions for Simulation #4 are very successful. The selectivity of A in Reaction I increases to 2.28 moles of A per mole of C. The new operating policies for the distillations result in high fractional recoveries of A and E as well as high purity. The magnitude of each cost component is the lowest obtained for any of the first four simulations. Total cost savings of 21.7 percent are realized over the results obtained in Simulation #3.

A few minor changes are made to Simulation #4 to generate Simulation #5. First, a rapid quench is included at the tail-end of Reaction I. This quench has essentially no effect on the final compositions for Reaction I. The use of jacket cooling shortens the time requirement for Reaction I. The other change involves merging Reactions I and II into a single stage. This change increases the average rate for the two tasks by eliminating a

transfer time. The result is a decrease in total equipment costs. These changes reduce the total costs by 6.0 percent compared to Simulation #4.

Simulation #6 has some minor changes in the cut locations for the distillations. These changes did not have a beneficial effect on the total cost. Although a slightly greater recovery of A and E is obtained in Distillation I, the reduction in purity increases waste treatment costs enough to outweigh the savings in raw materials and utilities costs. The overall cost increases by 1.3 percent over the results obtained in Simulation #5.

No additional iterations are done after Simulation #6 for two main reasons. First, diminishing returns are being obtained with each additional simulation. In fact, Simulation #6 has a total cost increase. Second, the performance measures indicate that the magnitude of further improvements should be small. The processing conditions for Simulation #5 result in high selectivity, high fractional recoveries, and high product purities. Also, the average processing rates for the four stages are extremely well balanced. Although slight additional improvements could probably be obtained with additional simulations, the number of simulations will increase significantly as finer trade-offs are explored.

7.8 Final Design

This section describes in detail all the operating conditions and the process structure for the final design (Simulation #5). The rationale behind the final choices for values for the operating variables and processing times is also provided. Some of the more important operational issues are discussed, as well as areas for possible further improvement.

Reactions I and II are both carried out in the first stage of processing in the final design. The detailed operating conditions for Reactions I and II are shown in Table 7-12. These two tasks are merged to increase the average production rate. Two reactors (500 and 1000 gallons) are operated in parallel at stage 1 to carry out the two reactions. The stage cycle time is 13.1 hours, including transfer times.

Reaction I uses a feeding strategy to promote the formation of A over C. By feeding R2 over time, the concentrations of R2 and I1 are kept low while R1 is relatively high. This situation increases the relative reaction rates for producing product A. Jacket cooling is used during the first part of the reaction in an attempt keep the temperature down in order to increases the selectivity of A. After all of the R2 is fed, the jacket cooling is turned off, and the temperature goes rapidly to the boiling point. The higher temperature greatly increases the reaction rate, allowing the reaction to satisfy the conversion

TABLE 7-12.
OPERATING CONDITIONS FOR REACTIONS I AND II FOR FINAL DESIGN

TASKS	DECISIONS	VALUES
REACTION I	R1 / R2 Ratio	2.0 mol R1 / mol R2
	S / R2 Ratio	0.16 liters S / mol R2
	CAT / R2 Ratio	3.30 ml CAT / mol R2
	Feed Temperature	80 °C for initial charge
		60 °C for remaining feed
	Addition Strategy	Charge: 100% R1,S; 10% R2; 38.5% CAT
		0 < t < 5 hr: Remaining 90% R2
		0 < t < 7.8 hr: Remaining 61.5% CAT
	Reaction Time	8 hr
	Jacket Duty	0 < t < 3.2 hr: 25,000 kcal/hr cooling
		3.2 < t < 5.0 hr: 13,000 kcal/hr cooling
		5.0 < t < 7.8 hr: 0 kcal/hr
		7.8 < t < 8.0 hr: 100,000 kcal/hr cooling
	Maximum Reflux Condenser Duty	100 kcal/ hr / gal reaction material
REACTION II	R3 / C Ratio	3.0 mol R3 / mol C
	Reaction Time	3.58 hr (99% conversion of C)
	Temperature	75 °C

specification on R2 within a reasonable amount of time. However, the high selectivity to A is lost during this second phase of the reaction. A final selectivity of 2.28 moles of A per mole of C is obtained.

Because of the exothermic and batch aspects of the reaction, temperature control is very difficult. Too much cooling quickly quenches the reaction; too little cooling allows a rapid temperature increase to the boiling point. An extremely rapid temperature increase could pose a safety hazard for two reasons. First, sudden boiling could cause an excessive pressure build-up in the reactor. Second, if the condensing system is suddenly overloaded, uncondensed toxic vapor might be released to the atmosphere. The feeding policy helps regulate the reaction rate and thus the exothermic heat release, making temperature control easier. The determination of the reaction jacket cooling profile and the reactant feeding strategy are two key areas for additional optimization work for Reaction I.

A feed ratio of 3.0 moles of R3 per mole of C is selected for Reaction II to provide a conversion of 99 percent in 3.58 hours. Lower feed ratios required too long to reach this high conversion, which exceeds the constraint specification of 98 percent. Because R3 is so inexpensive and such a high conversion of C is required, there is little opportunity or need to explore operating trade-offs in great detail for Reaction II.

Distillations I and II are carried out in the second stage. Table 7-13 shows the operating policies for these two tasks. The 1000 gallon still with a constant vapor rate of 20 kmol per hour is used at stage 2 to carry out the two distillations. The stage cycle time is 8.27 hours, including all total reflux, heating, cooling, and transfer times.

Multiple reflux ratios per cut are employed in order to increase the sharpness of the splits and reduce the amount of product losses. Low reflux ratios are used to quickly take most of a cut with a minimum utility cost. Higher reflux ratios are then used to provide better sharpness in the vicinity of the cut. Lower pressures in the second half of Distillation I and during Distillation II also increase the sharpness of the splits obtained. The resulting cost increases are not large, mainly because of the form of the vacuum cost function. Overall, 97.06 percent (by moles) of the A and E fed to Distillations I and II is recovered and sent on to Reaction III.

Reaction III occurs in a 500 gallon reactor at stage 3 in the final process. The operating conditions for Reaction III are shown in Table 7-14. Since there are no cost terms that include the temperature of Reaction III, the maximum temperature of 95 °C is

TABLE 7-13.
OPERATING CONDITIONS FOR DISTILLATIONS I AND II FOR FINAL DESIGN

TASKS	DECISIONS	VALUES
DISTILLATION I	Reflux Ratio	1.0 for Operating Step #1 6.0 for Operating Step #2
	Pressure	760 mm Hg for Operating Step #1 2 mm Hg for Operating Step #2
	Cut Location	Mole Fraction E in distillate reaches 0.01 for Operating Step #1 Mole Fraction E in distillate reaches 0.05 for Operating Step #2
DISTILLATION II	Reflux Ratio	2.0
	Pressure	2. mm Hg
	Cut Location	Pot Temperature reaches 135 °C

TABLE 7-14.
OPERATING CONDITIONS FOR REACTION III AND DISTILLATION III FOR FINAL DESIGN

TASK	DECISION	VALUE
REACTION III	R4 / E Ratio	25 mol R4 / mol E
	Temperature	95 °C
	Reaction Time	6.01 hr (93.5% conversion of E)
DISTILLATION III	Reflux Ratio	1. for Operating Step #1 5. for Operating Step #2
	Pressure	760. mm Hg for Operating Steps #1, #2
	Cut Location	Purity of A & D in pot of 85% (mass) for for Operating Step #1 Purity of A & D in pot of 99% (mass) for for Operating Step #2

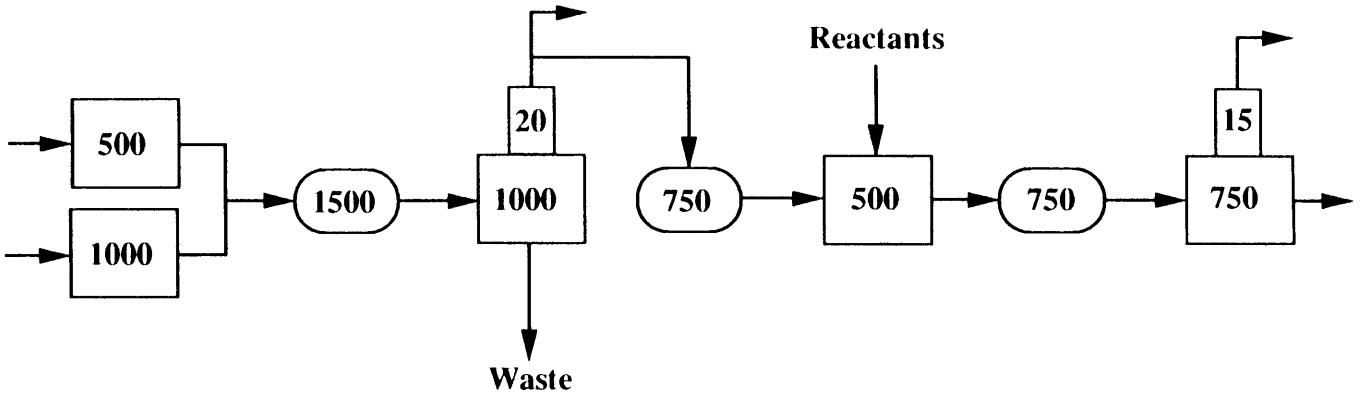
used in order to maximize the reaction rate. The total stage cycle time is 7.0 hours. Because the kinetic rate expression is independent of R4 (assuming R4 is in large excess), the R4 to E ratio is set at the constraint value of 25. Although R4 is very inexpensive, the waste treatment cost for the overheads from Distillation III that contains R4 is quite high. Minimizing the amount of excess R4 minimizes the raw materials and waste treatment costs. Given the existing constraint on the R4 to E ratio, the only further optimization possible for Reaction III would involve the processing time, i.e. the reaction conversion.

Distillation III occurs at stage 4 and completes the final purification of products A and D. A 750 gallon still with a vapor rate of 15 kmol per hour is used at this stage. The operating conditions for Distillation III are also shown in Table 7-14. The total cycle time for this stage is 10.38 hours. This separation removes S, R3, R4, and E overhead, leaving 99 percent pure A and D (by mass) in the bottoms. As in Distillations I and II, multiple reflux ratios are used to improve the sharpness of the splits. The final ratio of A to D is 3.367 (by mass).

Figure 7-3 shows a process flowsheet for the final design. The four stages are very well balanced in terms of average rates. The process uses intermediate storage between each pair of stages. Table 7-15 summarizes the results for the final design. Unlike the original base case, raw materials are now the largest cost factor, accounting for 49.4 percent of the total operating costs. The final process also satisfies the campaign time horizon constraint quite easily, needing only 30.1 days to finish.

Significant cost reductions have been achieved in going from the original base design to the final design. The overall cost has been reduced from over $6.3 million to just under $2.5 million, and the campaign time has been shortened by over 85 percent. These exceptionally large improvements exaggerate the benefits of doing performance and structure optimization because the base case conditions were so poor. However, significant cost reductions (26.4 percent) are obtained even when comparing the final results to the results for Simulation #3, which are much better than the base case. Thus, even when reasonably good base case values are available from laboratory and pilot plant studies, the benefits of optimizing the performance and structure of the overall process should still be sizeable.

Interactions between the performance and structure have been observed during the course of the evolutionary method. The best structure is different for each different set of processing conditions. Doing either structure of performance optimization alone would not



	Reaction I, II	Distillation I, II	Reaction III	Distilation III
Average Rate (kg/hr)	188.2	198.3	191.9	194.3

FIGURE 7-3. FINAL PROCESS FLOWSHEET

TABLE 7-15.
RESULTS FOR FINAL DESIGN

PERFORMANCE MEASURES	RESULTS
Reaction I:	
Selectivity (mol A / mol C)	2.282
% Conversion R2 to A or C	99.74 %
Distillation I:	
% Recovery of A and E (by mol)	97.43 %
Mole Fraction (A + E) in product stream	0.9796
Distillation II:	
% Recovery of A and E (by mol)	99.62 %
Mole Fraction (A + E) in product stream	0.9968
Overall Molar Yield: (A + 2D) / R2	94.69 %
Bottleneck Average Production Rate	188.2 kg/hr

ECONOMIC MEASURES	COSTS
Raw Materials Costs	$ 1216.2 K
Waste Treatment Costs	$ 121.1 K
Utilities Costs	$ 799.0 K
Equipment Usage Costs	$ 280.6 K
Intermediate Storage Costs	$ 43.4 K
TOTAL COSTS	$ 2460.2 K
CAMPAIGN TIME	30.1 days

have generated nearly as good a final design. This point is illustrated by Table 7-16 which considers six combinations of performance conditions and process structures. Two performance conditions (base case and final design) and three structures (base case, base case after structural optimization only, and final design) are used. The benefits of optimizing process performance and structure together are clear.

Any additional optimization efforts should focus on the operation of Reaction I. The lower bound on the raw materials cost is \$1004.3K, which occurs with complete conversion of R2 to A. Further work on increasing the selectivity without lengthening the reaction time could reduce this \$211.3K gap between the actual and ideal raw materials costs. Increasing the selectivity of A also tends to decrease waste treatment and utilities costs as well.

Some minor cost reductions might be obtained by doing additional fine tuning of the column operating conditions. In particular, the use of smoothly varying reflux ratio profiles would eliminate the need for extra total reflux periods within a cut. Optimized reflux ratio profiles could speed up column operations and reduce utilities costs. However, the magnitudes of these potential savings are expected to be much less than those possible by improving the operation of Reaction I.

7.9 Concluding Remarks

This chapter has considered the design of a batch process of industrial size and complexity. Performance issues have been included to demonstrate their impact on the design. A number of important conclusions can be drawn from this design study regarding the incorporation of process performance effects and the use of more detailed process models in the design and optimization of batch processes.

This case study illustrates the fact that larger industrial-like batch process design problems have the same types of general performance trade-offs that were shown for the simple two stage example problems discussed in Chapter 4. Both process intensity and performance load trade-offs are observed in the case study problem. These trade-offs have significant effects on the overall economics of the process.

The benefits of using more-detailed process models are clearly shown in this design problem. Process models have been used to explore performance trade-offs in order to improve the overall process. These models are important tools because local changes in

TABLE 7-16.
INTERACTIONS OF PERFORMANCE AND STRUCTURE DECISIONS ON OVERALL COST

PROCESS STRUCTURE	PERFORMANCE CONDITIONS	
	Base Case Conditions	Final Design Conditions
Original Base Case	$ 6312.5 K	$ 2874.1 K
Improved Base Case	$ 5255.3 K	$ 2683.6 K
Final Design	$ 6501.2 K	$ 2460.2 K

processing conditions do not always result in global improvements. Process models facilitate the use of an overall systems approach during design.

Although the mathematical programming solution procedures discussed in Chapter 4 are not applied to this problem because of excessive computation times, the use of systematic improvement methods does result in significant improvements to the process, both in terms of operating cost and production rate. The advantages of considering both performance and structure issues together at the design stage are also demonstrated.

The case study problem also serves as a testing ground for the optimization methods described in previous chapters. The issues of computational speed-ups and the need for good short-cut models are highlighted by this problem. The evolutionary approach used for this problem circumvented the need to do a large number of time consuming simulations. However, the beneficial effects of using more detailed process models to "optimize" the performance of a batch process have been retained. Further work is obviously required on ways to speed up mathematical programming approaches through the use of approximate methods or short-cut models. Without advances on these fronts, mathematical programming optimization techniques will not be able to handle batch performance optimization problems in reasonable computation times for large industrial-scale batch processes.

The solution procedures used for this design problem illustrated the benefits of applying approximate methods, particularly as regards the incorporation of structure issues. The approximate local search procedure worked very well on this problem while requiring very little computational effort (on the order of one cpu minute on a DEC Microvax). Although exact mathematical programming methods could have been used to obtain "optimal" results for various structure aspects, the near-optimal solutions obtained using these approximate methods were quite satisfactory. The approximate methods also required less effort in formulating and solving the particular problem. Further work could be done to incorporate intermediate storage and task assignment features directly into the local search procedure.

The success of these approximate methods (evolutionary approach, local search procedure) suggests an interesting extension. Perhaps suitable approximate methods could be used during batch process design to avoid the computational burden of solving these large, complicated mixed integer nonlinear mathematical programming problems. Approximate methods, which are usually based on some relatively straight-forward

engineering insights, also have the attractive feature of being easier for the practicing engineer to understand and implement in a reasonable (i.e., shorter) period of time. Approximate methods could be used to identify key processing and cost trade-offs and to get near-optimal solutions. Further refinement of critical decisions could be made with more exact tools if required. The uncertainties inherent in the process development and design atmosphere will often make near-optimal solutions good enough in many cases.

The use of approximate methods should not be construed to mean that process optimization (performance and/or structure) is not important during batch process design. One characteristic of batch processes is that there are many feasible designs. However, the scope for optimization is often large. Thus, efforts should be made to determine the processing conditions and structure that provide the best results. Only by comparing processing alternatives that are near their best can the best overall design be developed.

Chapter 8

SUMMARY AND AREAS FOR FUTURE WORK

This study represented a first attempt at including process performance issues in the optimal design of batch processes. Choices of operating variables and processing times were shown to have a significant impact on the quality of the final designs obtained for both simple example problems and an industrial-sized case study. The benefits of using more detailed process models to explore performance trade-offs were clearly shown. Mathematical programming solution procedures were used on small performance subproblems, but long computation times discourage the use of such methods for larger industrial-sized problems unless significant computational speed-ups are implemented.

The general environment of a multipurpose plant was shown to have an effect on the "optimal" operating conditions for producing a new product that is being incorporated into the product slate. A scheme for allocating fixed costs was devised in to reduce the complexity of the design problem while retaining the effects of the overall plant on the design of the new process. The use of existing equipment introduced combinatorial aspects to the optimization problem. Although the use of MINLP methods for the structure subproblem would provide global optimum solutions, an approximate method using local search techniques was quite successful in obtaining near-optimal solutions in a fraction of the computation time needed by MINLP approaches.

Interactions between process performance and the selection of equipment units were observed for both simple example problems and the more complex case study problem. These two sets of decisions cannot be made independently. Solution methods for the combined performance, structure optimization problem will need to either simultaneously handle both sets of decisions or properly coordinate the subproblems in order to obtain an overall optimum solution. The size and characteristics of the resulting MINLP discourage simultaneous solution. A decomposition strategy has been proposed, but further work is required to determine the best nesting arrangement of the performance and structure subproblems.

The large scope for optimization expected for most industrial-sized batch process design problems makes it unlikely that an exact optimum solution can be be obtained in a

reasonable amount of computation time. However, the large improvements shown in the case study problem clearly illustrate the great potential for process improvements by considering process performance in some type of systematic optimization approach. Approximate optimization strategies, such as the LS method and the evolutionary approach used for the case study problem, figure to play a prominent role in future efforts for optimization of both process performance and structure during the design of batch processes. The key issue for these approximate methods will be the trade-off between the computational load and the quality of solution.

Opportunities for future work focus on three areas. First, additional work is required on developing speed-ups, such as short-cut models or optimization strategies that employ reduced models, for use in the optimal design of batch processes. If these speed-ups could reduce the computational load enough, the use of more rigorous optimization approaches might be practical. A second area for additional work involves extending the use of the LS method to handle intermediate storage and task to stage assignment effects. Finally, additional work on the POSI nesting arrangement is required to identify the best algorithms for the outer loop and to determine the behavior of the method on larger problems.

REFERENCES

Aggarwal, A., Ciric, A.R., and Floudas, C.A., "Global Optimum Search for a Class of Non-Convex NLP and MINLP Problems," AIChE National Meeting, Washington, DC, (1988).

Aspen Technology, Inc., *BATCHFRAC User Manual, Aspen Technology, Inc.*, Cambridge, Massachusetts, (1987).

Aspen Technology, Inc., *ASPEN PLUS User Guide, Aspen Technology, Inc.*, Cambridge, Massachusetts, (1988).

Bamopoulos, G., Bacher, S., and Pan, S.H., "Computer-aided Design of Batch Chemical Processes," Paper presented at the 184th National ACS Meeting., Kansas City, MO, (Sept. 1981).

Batch Process Technology, Promotional Literature, (1987).

Birewar, D.B., and Grossmann, I.E., "Incorporating Scheduling in the Optimal Design of Multiproduct Batch Plants," AIChE Annual Meeting, New York, NY, (1987).

Birewar, D.B., and Grossmann, I.E., "Incorporating Scheduling in the Optimal Design of Multiproduct Batch Plants," *Comp. and Chem. Eng.*, **13**(1/2), 141 (1989a).

Birewar, D.B., and Grossmann, I.E., "Simultaneous Task Assignment and Scheduling in the Design of Multiproduct Batch Plants," AIChE Annual Meeting, San Francisco, CA, (1989b).

Boston, J.F., and Britt, H.I., "A Radically Different Formulation of the Single-Stage Flash Problem," *Comp. Chem. Eng.*, **41**, 109 (1978).

Boston, J.F., Britt, H.I., Jirapongphan, S., and Shah, V.B., "An Advanced System for the Simulation of Batch Distillation Operations," *Foundations of Computer-Aided Chemical Process Design, Vol. 2*, eds. R.S.H. Mah and W. Seider, Engineering Foundation, New York, 203 (1981).

Brannock, N.F., Verneuil, V.S., and Wang, Y.L., "PROCESS Simulation Program - A Comprehensive Flowsheeting Tool for Chemical Engineers," *Comp. and Chem. Eng.*, **3**, 329 (1979).

Cerda, J., Vicente, M., Gutierrez, J.M., Esplugas, S., and Mata, J., "A New Methodology for the Optimal Design and Production Schedule of Multipurpose Batch Plants," *Ind. Eng. Chem. Res.*, **28**, 988 (1989).

Chem. Share Corp., Guide to Solving Process Engineering Problems by Simulation (Users Manual), Chem. Share Corp., Houston, Texas, (1979).

Clark, S.M., and Kuriyan, K., "BATCHES - Simulation Software for Managing Simicontinous and Batch Processes," Paper No. 32f, AIChE National Meeting, Houston, (April 1989).

Clayton, R. ed., "Cost Reductions on a Grain Whiskey Distillery Identified by a Process Integration Study at Long John International Ltd.," Report RD/4 Energy Efficiency Office, U. K. (1987a).

Clayton, R. ed., "Cost Reductions on an Edible Oil Refinery Identified by a Process Intergration Study at Vanden Bergs & Jungens Ltd.," Report RD/14 Energy Efficiency Office, U. K. (1987b).

Clayton, R. ed., "Cost Reductions on a Synthetic Resins Site Identified by a Process Integration Study at Cray Valley Products," Report RD/30 Energy Efficiency Office, U. K. (1987c).

Coulman, G.A., "An Algorithm for Optimal Scheduling and a Revised Formulation of Batch Plant Design," *Ind. Eng. Chem. Res.*, **28**, 553 (1989).

Cuthrell, J.E., and Biegler, L.T., "Simultaneous Solution and Optimization of Process Flowsheets with Differential Equation Models," *I. Chem E. Symp. Ser. #92*, **25** (1985).

Cuthrell, J.E., and Biegler, L.T., "Simultaneous Solution and Optimization of Process Flowsheets with Differential Equation Models," *Chem. Eng. Res. Des.*, **64**, 341 (1986).

Das, H., Cummings, P.T., and LeVan, M.D., "Application of Simulated Annealing to Scheduling of Serial Multiproduct Batch Processes," AIChE Annual Meeing, San Francisco, Ca (1989).

Embury, M.C., Reklaitis, G.V., and Woods, J.M., "Simulation of the Operation of a Staged Multi-Product Process with Limited Interstage Storage Buffers," Proc. 9th Hawaii Conf. Sys. Sci., 348 (1976).

Embury, M.C., Reklaitis, G.V., and Woods, J.M., "Scheduling and Simulation of a Staged Semi-Continuous Multi-Product Process," Proceedings, 2nd Pacific Area Chemical Engineering Congress, Denver, CO, (Aug.1977).

Espuna, A., Lazaro, M., Martinez, J.M., and Puigjaner, L., "Efficient and Simplified Solution to the Predesign Problem of Multiproduct Plants," AIChE Annual Meeting, New York, NY, (1987).

Espuna, A., Lazaro, M., Martinez, J.M., and Puigjaner, L., "An Efficient and Simplified Solution to the Predesign Problem of Multiproduct Plants," *Comp. and Chem. Eng.*, **13**(1/2), 163 (1989)

Evans, L.B., Trevino-Lozano, R.A., Britt, H.I., and Boston, J.F., "Simultaneous Modular Process Simulation and Optimization," *I. Chem E. Symp. Ser. #92*, **25** (1985).

Faqir, N.M., and Karimi, I.A., "Optimal Design of Batch Plants with Single Production Routes," AIChE Annual Meeting, Washington, D.C., (1988).

Faqir, N.M., and Karimi, I.A., "Optimal Design of Batch Plants with Single Production Routes," *Ind. Eng. Chem. Res.*, **28**, 1191 (1989a).

Faqir, N.M., and Karimi, I.A., "Design of Multipurpose Batch Plants with Multiple Production Routes," presented at Foundations of Computer Aided Process Design, Snowmass, CO, July 9-14, (1989b).

Felder, R.M., "Simulation - A Tool for Optimizing Batch Process Production," *Chemical Engineering*, April 18, 79 (1983a).

Felder, R.M., Kester, P.M., and McConney, J.M., "Simulation / Optimization of a Specialties Plant," *CEP*, **June**, 84 (1983b).

Fisher, W.R., Doherty, M.F., and Douglas, J.M., "Evaluating Significant Economic Trade-offs for Process Design and Steady-State Control Optimization Problems," *AIChE J.*, **31**(9), 1538 (1985).

Flatz, W., "Equipment Sizing for Multiproduct Plants," *Chemical Engineering*, **87**, 71, (1980).

Flatz, W., "Sizing Equipment for Multiproduct Plants," *Chemical Engineering*, **88**, 105, (1981).

Fletcher, R., *Practical Methods of Optimization, Vol I, Unconstrained Optimization*, John Wiley & Sons, Inc., New York (1981).

Fletcher, R., *Practical Methods of Optimization, Vol II, Constrained Optimization*, John Wiley & Sons, Inc., New York (1981).

Fruit, W.M., Reklaitis, G.V., and Woods, J.M., *Chem. Eng. J.* , **8**, 199 (1974).

Ganesh, N., and Biegler, L.T., "A Robust Technique for Process Flowsheet Optimization Using Simplified Model Approximations," *Comp. Chem. Eng.*, **11**(6), 553 (1987).

Grossmann, I.E., and Sargent, R.W.H., "Optimum Design of Multipurpose Chemical Plants," *Ind. Eng. Chem. Process Des. Dev.*, **18**(2), 343 (1979).

Hainsworth, J., "Optimal Design for Polymer Processes," *The Chemical Engineer*, **January**,14 (1984).

Hatipoglu, T., and Rippin, D.W.T., "The Effects of Optimal Temperature and Feed Addition Rate Profiles on Batch Reactions and Sequences of Batch Operations - The Development of a Catalogue of Attainable Regions," Institution of Chemical Engineers Symposium Series No. 87, Pergamon Press, 447 (1984).

Hellinckx, L.J., and Rijckaert, M.J., "Minimization of Capital Investment for Batch Processes," *Ind. Eng. Chem. Process Des. Dev.*, **10**(3), 422 (1971).

Hooke, R. and Jeeves, T.A., *J. Assoc. Computer Machines*, **8**, 212 (1961).

Imai, M., and Nishida, N., "New Procedure Generating Suboptimal Configurations to the Optimal Design of Multipurpose Batch Plants," *Ind. Eng. Chem. Process Des. Dev.*, **23**(2), 845 (1984).

Janicke, W., "Optimal Assignment of Products to Equipment Items in Multipurpose Batch Chemical Plants," *Comp. and Chem. Eng.*, **11**(3), 303 (1987)

Jirapongphan, S., *Simultaneous Modular Convergence Concept in Process Flowsheet Optimization,* Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA (1980).

Jirapongphan, S., Boston, J.F., Britt, H.E., and Evans, L.B., "A Nonlinear Simultaneous Modular Algorithm for Process Flowsheeting Optimization," AIChE Annual Meeting, Chicago, IL (1980).

Joglekar, G.S., and Reklaitis, G.V., "A Simulator for Batch and Semicontinuous Processes," *Comp. and Chem. Eng.*, **8**(6), 315 (1984)

Karimi, I.A., and Reklaitis, G.V., "Optimal Selection of Intermediate Storage Tank Capacity in a Periodic Batch/Semicontinuous Process," *AIChE J.*, **29**(4), 588 (1983).

Karimi, I.A., and Reklaitis, G.V., "Intermediate Storage in Noncontinous Processes Involving Stages of Parallel Units," *AIChE J.*, **31**(1), 44 (1985a).

Karimi, I.A., and Reklaitis, G.V., "Deterministic Variability Analysis for Intermediate Storage in Noncontinuous Processes: Part I: Allowability Conditions," *AIChE J.*, **31**(9), 1516 (1985b).

Karimi, I.A., and Reklaitis, G.V., "Deterministic Variability Analysis for Intermediate Storage in Noncontinuous Processes: Part II: Storage Sizing for Serial Systems," *AIChE J.*, **31**(9), 1528 (1985c).

Ketner, S.E., "Minimize Batch Equipment Cost," *Chemical Engineering*, **6/7**, Aug. 22, 121 (1960).

Kiraly, L.M., Friedler, F., and Szoboszlai, "Optimal Design of Multipurpose Batch Chemical Plants," Proceedings of CHEMDATA 88, Gothenburg, Sweden, 355, (1988).

Klossner, J., and Rippin, D.W.T., "Combinatorial Problems in the Design of Multiproduct Batch Plants -Extension to Multiplant and Partly Parallel Operation," AIChE Annual Meeting, San Francisco, CA, (1984).

Knopf, F.C., Okos, M.R., and Reklaitis, G.V., "Optimal Design of Batch/Semicontinuous Processes," *Ind. Eng. Chem. Process Des. Dev.*, **21**(1), 79 (1982).

Knopf, F.C., Reklaitis, G.V., and Okos, M.R., "Design of Regenerative Noncontinuous Processes via Simulation," *Ind. Eng. Chem. Process Des. Dev.*, **23**(3), 522 (1984).

Kocis, G.R., and Grossmann, I.E., "Relaxation Strategy for the Structural Optimization of Process Flowsheets," *Ind. Eng. Chem. Res.*, **26**, 1869 (1987).

Kocis, G.R., and Grossmann, I.E., "Global Optimization of Nonconvex Mixed Integer Nonlinear Programming (MINLP) Problems in Process Synthesis," *Ind. Eng. Chem. Res.*, **27**, 1407 (1988).

Kocis, G.R., and Grossmann, I.E., "Computational Experience with DICOPT Solving MINLP Problems in Process Systems Engineering," *Comp. and Chem. Eng.*, **13**(3), 307 (1989a).

Kocis, G.R., and Grossmann, I.E., "A Modelling and Decomposition Strategy for the MINLP Optimization of Process Flowsheets," *Comp. and Chem. Eng.*, **13**(7), 797 (1989b).

Kuriyan, K., Reklaitis, G.V., and Joglekar, G.S., "Multiproduct Plant Scheduling Studies Using BOSS," CIChE-AIChE Symposium on Modern Chemical Engineering Technology, Taipei, September 16-19, (1986)

Lazaro, M.,and Puigjaner, L., "Simulation and Optimization of Multi-Product Plants for Batch and Semi-Batch Operation," I. Chem. E. Symp. Series No. 92, 209, (1985).

Linnhoff, B., "Process Integration of Batch Processes," AIChE Annual Meeting, New York, NY, (1987).

Loonkar, Y.R., and Robinson, J.D., "Minimization of Capital Investment for Batch Processes," *Ind. Eng. Chem. Process Des. Dev.*, **9**(4), 625 (1970).

Lucia, A., "Global Methods for Chemical Process Optimization," AIChE Annual Meeting, New York, NY, (1987).

Malone, M.F., "Batch Sequencing by Simulated Annealing," AIChE Annual Meeting, San Francisco, CA (1989).

Mauderli, A., and Rippin, D.W.T., "Production Planning and Scheduling for Multi-Purpose Batch Chemical Plants," *Comp. and Chem. Eng.*, **3**, 199 (1979).

Mauderli, A., and Rippin, D.W.T., "Scheduling Production for Multi-Purpose Batch Plants: The Batchman Program," *Comp. Eng. Prog.*, **76**(4), 37 (1980).

Modi, A.K., and Karimi, I.A., "Design of Multiproduct Batch Processes with Finite Intermediate Storage," AIChE Annual Meeting, New York, NY, (1987).

Modi, A.K., and Karimi, I.A., "Design of Multiproduct Batch Processes with Finite Intermediate Storage," *Comp. and Chem. Eng.*, **13**(1/2), 127 (1989).

Musier, R.F.H., *Planning and Scheduling of Batch and Semi-Continuous Chemical Processes*, Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA , (May 1989).

Newberger, M.R., American Cyanamid Co., unpublished work, (1969).

Numerical Algorithms Group, NAG FORTRAN Library Manual, Mark 11, Numerical Algorithms Group, Oxford (1984).

Overturf, B.W., Reklaitis, G.V., and Woods, J.M., "GASP IV and the Simulation of Batch/Semicontinuous Operations: Single Train Process," *Ind. Eng. Chem. Process Des. Dev.*, **17**(2), 161 (1978a).

Overturf, B.W., Reklaitis, G.V., and Woods, J.M., "GASP IV and the Simulation of Batch/Semicontinuous Operations: Parallel Train Process," *Ind. Eng. Chem. Process Des. Dev.*, **17**(2), 166 (1978b).

Papadimitriou, C.H., and Steiglitz, K., *Combinatorial Optimization: Algorithms and Complexity*, Englewood Cliffs, NJ: Prentice-Hall, Inc. (1982).

Papageorgaki, S., and Reklaitis, G.V., "A Decomposition Approach to the Design of Multipurpose Batch Plants," AIChE Annual Meeting, San Francisco, CA, (1989).

Parakrama, R., "Improving Batch Chemical Processes," *The Chemical Engineer*, **September**, 24 (1985).

Pike, R.W., *Optimization for Engineering Systems*, Van Nostrand Reinhold Company, New York (1986).

Reinhart, H.J., and Rippin, D.W.T., "The Design of Flexible Batch Chemical Plants," AIChE Annual Meeting, New Orleans, LA, (1986).

Reinhart, H.J., and Rippin, D.W.T., "Design of Flexible Multi-Product Plants: A New Procedure for Optimal Sizing Under Uncertainty," AIChE Annual Meeting, New York, NY, (1987).

Reklaitis, G.V., "Review of Scheduling of Process Operations," *AIChE Symp. Ser. No. 214*, **78**, 119 (1982).

Reklaitis, G.V., "Progress and Issues in Computer Aided Batch Process Design," presented at Foundations of Computer Aided Process Design, Snowmass, CO, July 9-14, (1989).

Renfro, J.G., Morshedi, A.M., and Asbjornsen, O.A., "Simultaneous Optimization and Solution of Systems Described by Differential/Algebraic Equations," *Comput. Chem. Engng.*, **11**(5), 503 (1987).

Rich, S.H., and Prokopakis,G.J., "Scheduling and Sequencing of Batch Operations in a Multipurpose Plant," *Ind. Eng. Chem. Process Des. Dev.*, **25**(4), 979 (1986).

Rich, S.H., and Prokopakis,G.J., "Multiple Routings and Reaction Paths in Project Scheduling," *Ind. Eng. Chem. Res.*, **26**(9), 1940 (1987).

Rippin, D.W.T., "Review Paper: Simulation of Single- and Multiproduct Batch Chemical Plants for Optimal Design and Operation," *Comp. and Chem. Eng.*, **7**(3), 137 (1983b).

Rippin, D.W.T., "Design and Operation of Multiproduct and Multipurpose Batch Chemical Plants. An Analysis of Problem Structure," *Comp. and Chem. Eng.*, **7**(4), 463 (1983a).

Robinson, J.D., and Loonkar, Y.R., "Minimising Capital Investment for Multiproduct Batch Plants," *Process Technol. Intl.*, **17**(11), 861 (1972).

Rosen, E.M., and Pauls, A.C., "Computer Aided Chemical Process Design: The FLOWTRAN System," *Comp. and Chem. Eng.*, **1**, 11 (1977).

Sparrow, R.E., Rippin, D.W.T., and Forder, G.J., 'MULTIBATCH: A Computer Package for the Design of Multiproduct Batch Plants," *Chem. Eng.*, **289**, 520 (1974).

Sparrow, R.E., Forder, G.J., and Rippin, D.W.T., "The Choice of Equipment Sizes for Multiproduct Batch Plants. Heuristics vs. Branch and Bound," *Ind. Eng. Chem. Process Des. Dev.*, **14**(3), 198 (1975).

Suhami, I. and Mah, R.S.H., "Scheduling of Multipurpose Batch Plants with Product Precedence Constraints," *Proceedings of the Second FOCAPD Conference*, A.W. Westerberg and H.H. Chien, eds., American Institute of Chemical Engineers, New York (1984).

Suhami, I. and Mah, R.S.H., "Optimal Design of Multipurpose Batch Plants," *Ind. Eng. Chem. Process Des. Dev.*, **21**(1), 94 (1982).

Sundaram, S., and Evans, L.B., "Optimal Operation of Multicomponent Batch Distillation," AIChE Annual Meeting, San Francisco, CA (1989).

Sundaram, S., and Evans, L.B., "Development of a Fast, Accurate Short-Cut Model for Multicomponent Batch Distillation," submitted for presentation at AIChE Annual Meeting, Chicago, IL (1990).

Takamatsu, T., Hashimoto, I., and Hasebe, S., "Optimal Scheduling and Minimum Storage Tank Capacities in a Process System with Parallel Batch Units," *Comp. Chem. Eng.*, **3**, 185 (1979).

Takamatsu, T., Hashimoto, I., and Hasebe, S., "Optimal Design and Operation of a Batch Process with Intermediate Storage Tanks," *Ind. Eng. Cham. Process Des. Dev.*, **21**(3), 431 (1982).

Takamatsu, T., Hashimoto, I., Hasebe, S., and O'Shima, M., "Design of a Flexible Batch Process with Intermediate Storage Tanks," *Ind. Eng. Chem. Process Des. Dev.*, **23**(1), 40 (1984).

Trevino-Lozano, R.A., *Simultaneous Modular Concept in Chemical Process Simulation and Optimization,* Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA , (1985).

Vaselenak, J.A., "*Studies in the Optimal Design and Scheduling of Batch Processing Plants,*" Ph.D. Dissertation, Carnegie-Mellon University, (1985).

Vaselenak, J.A., Grossmann, I.E., and Westerberg, A.W., "Optimal Retrofit Design of Multi-product Batch Plants," *Ind. Eng. Chem. Res.*, **26**, 718 (1987a).

Vaselenak, J.A., Grossmann, I.E., and Westerberg, A.W., "An Embedding Formulation for the Optimal Scheduling and Design of Multipurpose Batch Plants," *Ind. Eng. Chem. Res.*, **26**, 139 (1987b).

Wellons, H.S., and Reklaitis, G.V., "The Design of Multi-Product Batch Plants Under Uncertainty with Staged Expansion," AIChE Annual Meeting, New York, NY, (1987).

Wellons, H.S., and Reklaitis, G.V., "The Design of Multi-Product Batch PlantsUnder Uncertainty with Staged Expansion," *Comp. and Chem. Eng.*, **13**(1/2), 115 (1989)

Wellons, M.C., and Reklaitis, G.V., "Optimal Schedule Generation for a Single-Product Production Line, Part 1: Problem Formulation," AIChE Annual Meeting, New York, NY, (1987a).

Wellons, M.C., and Reklaitis, G.V., "Optimal Schedule Generation for a Single-Product Production Line, Part 2: Identification of Dominant Unique Path Sequences," AIChE Annual Meeting, New York, NY, (1987b).

Wellons, M.C., and Reklaitis, G.V., "Optimal Schedule Generation for a Single-Product Production Line - I. Problem Formulation," *Comp. and Chem. Eng.*, **13**(1/2), 201 (1989a).

Wellons, M.C., and Reklaitis, G.V., "Optimal Schedule Generation for a Single-Product Production Line - II. Identification of Dominant Unique Path Sequences," *Comp. and Chem. Eng.*, **13**(1/2), 213 (1989b).

Wiede, W. Jr., Kuriyan, K., and Reklaitis, G.V., "Discrete Variable Optimization Strategies for the Design of Multiproduct Processes," AIChE Annual Meeting, New Orleans, LA, (1981).

Wilson, J.A., "Dynamic Model Based Optimization in the Design of a Batch Process Involving Simultaneous Reaction and Distillation," Institution of Chemical Engineers Symposium Series No. 100, Pergamon Press, 163 (1987).

Yeh, N.C., and Reklaitis, G.V., "Synthesis and Sizing of Batch/Semicontinuous Processes," AIChE Annual Meeting, Chicago, IL, (1985).

Yeh, N.C., and Reklaitis, G.V., "Synthesis and Sizing of Batch/Semicontinuous Processes: Single Product Plants," *Comp. and Chem. Eng.*, **11**(6), 639 (1987).

Young, R.A., and Reklaitis, G.V., "Capacity Expansion Study of a Batch Production Line," *Ind. Eng. Chem. Res.*, **6**, 771 (1989).

APPENDICES

Appendix A.1: Model Equations for Example Problem #1

The model equation for the two stage process used in Section 4.5.1 are described here. The process consists of a batch reaction followed by a distillation column that operates with perfect splits.

Batch Reaction. The reaction produces product B and unwanted byproduct C from reactant A in the following first-order reactions:

$$A \rightarrow B \tag{A-1}$$

$$B \rightarrow C \tag{A-2}$$

Assuming pure A initially, the time dependent mole fractions are given by:

$$x_A = \exp(-k_1 t_r) \tag{A-3}$$

$$x_B = \frac{k_1}{k_2 - k_1}\left(\exp(-k_1 t_r) - \exp(-k_2 t_r)\right) \tag{A-4}$$

$$x_C = 1 - x_A - x_B \tag{A-5}$$

The reactor cycle time constraint is given by:

$$T_r \geq t_r + t_{cr} \tag{A-6}$$

Batch Distillation. The mixture of A, B, and C is fed to a batch distillation stage to separate the components. The column is assumed to generate perfect splits between the pairs of components. Reactant A has the highest relative volatility, so it is recovered first. Product B is taken as the second overhead fraction, leaving byproduct C in the pot. The perfect splits assumption causes the column operation time to be directly proportional to the amount of A and B in the feed to the column:

$$t_c = \frac{V_c C}{F_d} (x_A + x_B) \tag{A-7}$$

The column cycle time constraint is given by:

$$T_c \geq t_c + t_{cc} \tag{A-8}$$

Average Production Rate. The two stages are constrained to operate with the same average production rate in order to keep the size of the intermediate storage reasonable. The following average rate constrain must be satisfied.

$$\frac{V_r}{T_r} = \frac{V_c}{T_c} \tag{A-9}$$

Then, the campaign time (excluding end effects) required to produce a specified amount of product B is given by:

$$T_{tot} = \frac{B_{tot} T_r}{C V_r x_B} \tag{A-10}$$

The campaign must be completed within a given horizon time:

$$T_{tot} \leq T_{hor} \tag{A-11}$$

Cost Terms. The total production cost includes terms for raw materials, waste treatment, unit clean-outs, equipment rental, and column utilities.

$$\phi_{\text{raw materials}} = P_A \left(\frac{C V_r T_{tot}}{T_r}\right) \tag{A-12}$$

$$\phi_{\text{waste}} = (P_{rA} x_A + P_C x_C) \left(\frac{C V_r T_{tot}}{T_r}\right) \tag{A-13}$$

$$\phi_{\text{clean-out}} = \left(\frac{C_{clr}}{T_r} + \frac{C_{clc}}{T_c}\right) T_{tot} \tag{A-14}$$

$$\phi_{\text{equipment}} = (r_R + r_s + r_c) T_{tot} \tag{A-15}$$

$$\phi_{utilities} = \left(\frac{P_u F_d t_c T_{tot}}{T_c}\right) \quad \text{(A-16)}$$

Thus, the total cost for producing the campaign is:

$$\phi = \phi_{raw\ materials} + \phi_{waste} + \phi_{clean\text{-}out} + \phi_{equipment} + \phi_{utilities} \quad \text{(A-17)}$$

Dimensionless Formulation. The model equations can be put into a dimensionless form by scaling the reaction time and column cycle time by characteristic values. The characteristic reaction time is:

$$t_{r_{char}} = \frac{1}{k_1} \quad \text{(A-18)}$$

The characteristic column cycle time is the time to completely empty the still of all of its contents

$$T_{c\ char} = \frac{V_c C}{F_d} \quad \text{(A-19)}$$

Then, dimensionless decision variables are determined by:

$$T_r^* = \frac{t_r}{T_{r\ char}} \quad \text{(A-20)}$$

$$T_c^* = \frac{t_c}{T_{c\ char}} \quad \text{(A-21)}$$

The dimensionless form of the minimum cost performance subproblem is given below. dimensionless groups are defined in Table A-1.

$$\min \phi^* = \frac{(1 + N_A x_A + N_C x_C + N_u (1\text{-}x_C) + N_{clean} + N_{cap} T_c^*)}{x_B} \quad \text{(A-22)}$$

subject to

$$x_A = \exp(-t_r^*) \quad \text{(A-23)}$$

$$x_B = \frac{\exp(-t_r^*) - \exp(-N_{rxn}\, t_r^*)}{N_{rxn} - 1} \qquad \text{(A-24)}$$

$$x_C = 1 - x_A - x_B \qquad \text{(A-25)}$$

$$1 - \frac{N_{hor}\, T_c^*}{x_B} \geq 0 \qquad \text{(A-26)}$$

$$T_c^* - (x_A + x_B) - N_{chc} \geq 0 \qquad \text{(A-27)}$$

$$N_{rate}\, T_c^* - t_r^* - N_{chr} \geq 0 \qquad \text{(A-28)}$$

Table A-1 also provides the parameter values used for Example Problem #1 in Section 4.5.1.

Nomenclature for Appendix A.1.

B_{tot}	Total amount of product B required, mol.
C	Average concentration, mol / l.
C_{clc}	Average cleaning cost per column batch, $.
C_{clr}	Average cleaning cost per reactor batch, $.
F_d	Column distillate rate, mol / hr.
k_1	1st order reaction rate constant for A $\rightarrow$ B, hr $^{-1}$.
k_2	1st order reaction rate constant for B $\rightarrow$ C, hr $^{-1}$.
N_A	Dimensionless groups for recycle costs.
N_C	Dimensionless group for byproduct costs.
N_{cap}	Dimensionless group for equipment rental costs.
N_{chc}	Dimensionless group for column changeover time.
N_{chr}	Dimensionless group for reactor changeover time.
N_{clean}	Dimensionless group for cleaning costs.
N_{hor}	Dimensionless group for horizon constraint.
N_{rate}	Dimensionless group for relative production rates.
N_{rxn}	Dimensionless group for rate constant.
N_u	Dimensionless group for utility costs.
P_A	Cost of fresh feed A, $ / mol.

P_C	Cost / credit for byproduct C, \$ / mol.
P_{rA}	Disposal cost or recycle credit for A, \$ / mol.
P_u	Utility costs per mole of distillate, \$ / mol distillate.
r_C	Rental rate for the column, \$ / hr.
r_R	Rental rate for the reactor, \$ / hr.
r_s	Rental rate for intermediate storage, \$ / hr.
t_c	Column operation time, hr.
T_c	Column cycle time, hr.
T_c^*	Dimensionless column cycle time.
t_{cc}	Column changeover time, hr.
$T_{c\ char}$	Characteristic column cycle time, hr.
t_{cr}	Reactor changeover time, hr.
T_{hor}	Horizon time, hr.
t_r	Reactor operation time, hr.
$t_{r\ char}$	Characteristic reactor time, hr.
t_r^*	Dimensionless reactor operation time.
T_r	Reactor cycle time, hr.
T_{tot}	Total time required for campaign, hr.
V_c	Column still pot volume, l.
V_r	Reactor volume, l.
x_A	Mole fraction of A leaving reactor.
x_B	Mole fraction of B leaving reactor.
x_C	Mole fraction of C leaving reactor.
ϕ	Total operating costs, \$.
ϕ^*	Dimensionless total operating costs.
$\phi_{clean\text{-}out}$	Total cost for unit clean-outs, \$.
$\phi_{equipment}$	Total cost for equipment rental, \$.
$\phi_{raw\ materials}$	Total cost for raw materials, \$.
$\phi_{utilities}$	Total cost for column utilities, \$.
ϕ_{waste}	Total cost for waste treatment, \$.

Appendix A.2: Model Equations for Example Problem #1

The model equations for the two stage process used in Section 4.5.2 are described here. The process is very similar to the one described in Appendix A.1 with the exception of some minor changes described below.

Batch Reaction. Temperature dependent kinetics have been introduced into the reaction model. Arrhenius expressions relate the reaction rate constants to the reaction temperature

$$k_1 = pre_1 \exp\left[\frac{-E_1}{1.987\ Temp}\right] \tag{A-29}$$

$$k_2 = pre_2 \exp\left[\frac{-E_2}{1.987\ Temp}\right] \tag{A-30}$$

The remaining model equations for the batch reactor are given by Equations A-3 through A-6.

Batch Distillation. A perfect splits column is again used for the batch distillation. For this problem, a constant boil-up rate is specified rather than an effective distillate rate. However, these variables can be related if an effective reflux ratio for the separation is specified:

$$F_d = \frac{Vap}{R + 1} \tag{A-30}$$

The perfect splits assumption essentially means that a reflux ratio of R generates splits sharp enough to satisfy any necessary purity specifications. The remaining column model equations are given by Equations A-7 and A-8.

Average Production Rate. The equations involving the average rate are given by Equations A-9 through A-11.

Cost Model. The costs for this problem are calculated as shown below:

$$\phi_{raw\ materials} = \frac{P_A\ B_{tot}}{x_B} \tag{A-32}$$

$$\phi_{waste} = \frac{B_{tot}}{x_B} (P_{rA}\, x_A + P_C\, x_C) \qquad \text{(A-33)}$$

$$\phi_{clean\text{-}out} = \left(\frac{C_{clr}}{T_r} + \frac{C_{clc}}{T_c}\right) T_{tot} \qquad \text{(A-14)}$$

$$\phi_{equipment} = (r_r + r_c + C_s\,(V_r + V_c))\, T_{tot} \qquad \text{(A-34)}$$

$$\phi_{r\ util} = P_{ur}\, V_r\, (\,Temp - Temp_t\,) \frac{T_{tot}}{T_r} \qquad \text{(A-35a)}$$

$$\phi_{c\ util} = \frac{P_{uc}\, Vap\, t_c\, T_{tot}}{T_c} \qquad \text{(A-35b)}$$

$$\phi_{utilities} = \phi_{r\ util} + \phi_{c\ util} \qquad \text{(A-37)}$$

The total operating cost is given by Equation A-17. The values for the various parameters used for Example Problem #2 in Section 4.5.2 are shown in Table A-2.

TABLE A-2
PARAMETER VALUES FOR EXAMPLE PROBLEM #2

B_{tot}	200,000 mol	P_{ur}	\$ 0.012 / l K
C	4 mol / l	pre_1	2×10^{10} hr^{-1}
C_{clc}	\$ 150 / batch	pre_2	1×10^{10} hr^{-1}
C_{clr}	\$ 250 / batch	R	5.0
C_s	\$ 0.005 / hr / l	r_c	\$ 50 / hr
E_1	20,000 cal / mol K	r_R	\$ 45 / hr
E_2	20,000 cal / mol K	t_{cc}	6 hr
H	4,800 hr	t_{cr}	4 hr
P_A	\$ 1 / mol	$Temp_{ref}$	300 K
P_C	\$ 0.5 / mol	V_c	1,000 l
P_{rA}	\$ 0.15 / mol	V_r	2,000 l
P_{uc}	\$ 0.1125 / mol vaporized	Vap	1,200 mol / hr

Additional Nomenclature Introduced in Appendix A.2.

C_s	Cost coefficient for storage, \$ / hr-l.
E_1	Activation energy for A → B, cal / mol-K.
E_2	Activation energy for B → C, cal / mol-K.
P_{uc}	Column utility cost coefficient, \$ / mol vaporized.
P_{ur}	Reactor utility cost coefficient, \$ / l-K.
pre_1	Pre-exponential factor for A → B, hr^{-1}.
pre_2	Pre-exponential factor for B → C, hr^{-1}.
R	Column reflux ratio.
Temp	Reaction temperature, K.
$Temp_f$	Reactor feed temperature, K.
Vap	Column boil-up rate, mol / hr.

Appendix A.3: Example Problem #3

Three versions of the basic two stage process described in Appendix A.2 are solved in Section 4.5.3. The differences in these three versions are summarized here.

Analytic Model. The first problem involves exactly the two stage process model from Appendix A.2 with some slight changes in the problem parameter values. Specifically, the activation energies and pre-exponential factors are changed. The parameter values are given in Table A-3.

TABLE A-3
PARAMETER VALUES FOR EXAMPLE PROBLEM #3

B_{tot}	200,000 mol	P_{ur}	\$ 0.012 / l K
C	4 mol / l	pre_1	6×10^7 hr^{-1}
C_{clc}	\$ 150 / batch	pre_2	6×10^{12} hr^{-1}
C_{clr}	\$ 250 / batch	R	5.0
C_s	\$ 0.005 / hr / l	r_c	\$ 50 / hr
E_1	15,000 cal / mol K	r_R	\$ 45 / hr
E_2	25,000 cal / mol K	t_{cc}	6 hr
H	4,800 hr	t_{cr}	4 hr
P_A	\$ 1 / mol	$Temp_{ref}$	300 K
P_C	\$ 0.5 / mol	V_c	1,000 l
P_{rA}	\$ 0.15 / mol	V_r	2,000 l
P_{uc}	\$ 0.1125 / mol vaporized	Vap	1,200 mol / hr

Numerical Model. The second version of the process model involves replacing the analytic expressions for the mole fractions (Equations A-3 to A-5). Instead, the differential equations describing the rate of change of the mole fractions must be integrated numerically. The expressions given below assume no change in volume with reaction.

$$\frac{dx_A}{dt} = -k_1 x_A \qquad \text{(A-37)}$$

$$\frac{dx_B}{dt} = k_1 x_A - k_2 x_B \tag{A-38}$$

$$\frac{dx_C}{dt} = k_2 x_B \tag{A-39}$$

An explicit first-order Euler integration method is used to update the mole fractions:

$$x_A(t + \Delta t) = x_A(t) + \left(\frac{dx_A(t)}{dt}\right) \Delta t \tag{A-40}$$

$$x_B(t + \Delta t) = x_B(t) + \left(\frac{dx_B(t)}{dt}\right) \Delta t \tag{A-41}$$

Equations A-5 is used (rather than integrating Equation A-39) to insure that mole fractions sum to unity. Other than these changes, the process model uses the equations described in Appendix A.2 with the problem parameter values given in Table A-3.

Process Model with Sundaram Column Model. The third version of the problem involves replacing the perfect splits column model with the short-cut model developed by Sundaram and Evans (1990). The interested reader is referred to their work for details on the short-cut model equations. Essentially though, the column model takes a series of process inputs and returns a number of outputs. These relationships are described in general form by the following equations:

$$x_{Bp} = f_1 (x_A, x_B, x_C, \alpha_{AC}, \alpha_{BC}, R, N, cut_1, cut_2) \tag{A-42}$$

$$x_{Ar} = f_2 (x_A, x_B, x_C, \alpha_{AC}, \alpha_{BC}, R, N, cut_1, cut_2) \tag{A-43}$$

$$x_{Cw} = f_3 (x_A, x_B, x_C, \alpha_{AC}, \alpha_{BC}, R, N, cut_1, cut_2) \tag{A-44}$$

The final product cut must satisfy a purity specification.

$$x_{Bp} \geq x_{Bspec} \tag{A-45}$$

The column operating time is given by:

$$t_c = \frac{V_c\, C\, (R + 1)\, cut_2}{Vap} \tag{A-46}$$

The total campaign time is given by:

$$T_{tot} = \frac{B_{tot}\, T_r}{V_r\, (cut_2 - cut_1)\, C\, x_{Bp}} \tag{A-47}$$

The same general cost model is used as before, but two terms are modified slightly.

$$\phi_{raw\ matertials} = \frac{P_A\, B_{tot}}{V_r\, C\, x_{Bp}\, (cut_2 - cut_1)} \tag{A-48}$$

$$\phi_{waste} = \frac{(P_{rA}\, x_{Ar}\, cut_1 + P_C\, x_{Cw}\, (1 - cut_2))\, B_{tot}}{(cut_2 - cut_1)\, x_{Bp}} \tag{A-49}$$

The optimization problem involves minimizing Equation A-17. Equations A-29, A-30 and A-3 the A-6 describe the operation of the batch reactor. Column operations are described by Equations A-8, A-42 to A-44, and A-46. The average production rate is calculated using Equations A-9 and A-47. Finally, processing costs are determined using Equations A-14 through A-17, A-48, and A-49. Constraints on product purity (Equation A-45) and maximum campaign time (Equation A-11) must also be satisfied. The distillation column has 8 stages, and the relative volatilities are α_{AC}=5 and α_{BC}=2 for this problem. The product cut must have a mole fraction of B of at least 0.90.

Additional Nomenclature Introduced in Appendix A.3.

cut_1	Location of the first cut in the distillation , % distilled.
cut_1	Location of the second cut in the distillation , % distilled.
f_1	Nonlinear "function" representing the black box model equations that generate x_{Bp}.
f_2	Nonlinear "function" representing the black box model equations that generate x_{Ar}.
f_3	Nonlinear "function" representing the black box model equations that generate x_{Cw}.
N	Number of stages in distillation column.

t	Time during the reaction, hr.
Δt	Small increment of time during reaction, hr.
x_{Ar}	Mole fraction of A in 1st cut (recycle).
x_{Bp}	Mole fraction of B in 2nd cut (product).
x_{Bspec}	Specification on minimum product mole fraction.
x_{Cw}	Mole fraction of C left in still pot (waste).
α_{AC}	Relative volatility of A with respect to C.
α_{BC}	Relative volatility of B with respect to C.

Appendix A.4: Model Equations for One Stage Test Problem

The model equations for the one stage process described in Section 6.4.2 are summarized in the section. The process consists of a single reaction stage that produces product C. Reactants A and B are mixed, forming product C and byproduct D in the following reaction:

$$A + B \rightarrow C + D \tag{A-50}$$

The rate constant, k, is given by an Arrhenius expression

$$k = pre \ \exp\left[\frac{-E}{1.987 \ Temp}\right] \tag{A-51}$$

The differential equations for the kinetics are

$$\frac{dC_A}{dt} = -k \ C_A \ C_B \tag{A-52}$$

$$\frac{dC_B}{dt} = -k \ C_A \ C_B \tag{A-53}$$

$$\frac{dC_C}{dt} = k \ C_A \ C_B \tag{A-54}$$

$$\frac{dC_D}{dt} = k \ C_A \ C_B \tag{A-55}$$

These rate expressions can be integrated to obtain expressions for the composition profiles. Assuming the volume change of the reaction is small, ie., the total molar concentration is constant, a closed-form expression is obtained:

$$x_A(t) = \frac{(x_{Ao} - x_{Bo}) \ x_{Ao}}{x_{Ao} - x_{Bo} \exp\left[-(x_{Ao} - x_{Bo}) \ k \ C \ t_r\right]} \tag{A-56}$$

Equation A-56 holds only when $x_{Ao} \neq x_{Bo}$. If the initial mole fractions are equal, then Equation A-56 is replaced by:

$$x_A(t) = \frac{1}{2 + k\,C\,t_r} \tag{A-57}$$

Assuming only A and B initially, then the remaining compositions are given by:

$$x_B(t) = x_{Bo} - x_{Ao} + x_A \tag{A-58}$$

$$x_C(t) = \frac{(1 - x_A - x_B)}{2} \tag{A-59}$$

$$x_D(t) = \frac{(1 - x_A - x_B)}{2} \tag{A-60}$$

The reactor cycle involves filling the unit with material, pre-heating the reactants, carrying out the reaction, emptying the unit, cleaning-out or setting up for the next batch, and possibly some idle time. For a one stage process, idle time should always be zero unless there is a restriction on some resource. The reactor cycle time is given as:

$$T_r = t_{fill} + t_{heat} + t_r + t_{drain} + t_{clean} + t_{idle} \tag{A-61}$$

For this process, these operation time components are defined by:

$$t_{fill} = \frac{V_r}{R_{fill}} \tag{A-62}$$

$$t_{drain} = \frac{V_r}{R_{drain}} \tag{A-63}$$

$$t_{clean} = \text{constant} = 1 \text{ hour} \tag{A-64}$$

$$t_{heat} = \gamma\, V_r^{(1/3)} \ln\left[\frac{Temp_{ext} - Temp_i}{Temp_{ext} - Temp}\right] \tag{A-65}$$

For this problem, the filling and draining rates are assumed to be the same. The transfer rate is thus:

$$R_{trans} = R_{fill} = R_{drain} \tag{A-66}$$

The average rate for the process is the sum of the individual unit rates of all the units assigned to this reaction.

$$Rate_{tot} = Rate_{tot_r} = \sum_{i=1}^{k} \frac{y_{i_r} V_{r_i} C x_C}{T_{r_i}} \tag{A-67}$$

The campaign time is then:

$$T_{tot} = \frac{C_{tot}}{R_{tot}} \tag{A-68}$$

The total cost for producing C_{tot} units of product C consists of raw materials, waste treatment, unit clean-out costs, utilities usages, and equipment rental costs. These cost terms are shown below:

$$\phi_{raw\ materials} = (P_A x_{Ao} + P_B x_{Bo}) \frac{C_{tot}}{x_C} \tag{A-69}$$

$$\phi_{waste\ treatment} = (P_{rA} x_A + P_{rB} x_B + P_D x_D) \frac{C_{tot}}{x_C} \tag{A-70}$$

$$\phi_{utilities} = P_u C C_p (Temp - Temp_o) T_{tot} \sum_{i=1}^{n_r} \frac{V_{r_i} y_{i_r}}{T_{r_i}} \tag{A-71}$$

$$\phi_{clean\text{-}out} = \sum_{i=1}^{n_r} C_{clr} \frac{y_{ir} T_{tot}}{T_{r_i}} \tag{A-72}$$

$$\phi_{equipment} = (\sum_{i=1}^{n_r} y_{i_r} r_i) T_{tot} \tag{A-73}$$

The process must satisfy the following constraints:

$$T_{tot} \leq T_{hor} \tag{A-74}$$

$$Temp < Temp_{max} \quad \text{(A-75)}$$

$$x_C \geq x_{C\,min} \quad \text{(A-76)}$$

Five reactors are available for use with this process. Their characteristics are shown in Table A-4. Problem parameter values are specified in Table A-5.

TABLE A-4.
EQUIPMENT INVENTORY FOR ONE STAGE PROCESS

Reactor	Size (l)	Rental Cost ($/hr)
1	1000	56.80
2	2000	86.10
3	2500	99.60
4	3500	122.00
5	4000	130.00

TABLE A-5
PARAMETER VALUES FOR ONE STAGE PROBLEM

C	5 mol / l	P_u	0.008
C_{clr}	$ 916 / batch	pre	185,200 l / mol-hr
C_{tot}	500,000 mols C	R_{trans}	2,000 l / mol
E	12.9 kcal / mol	t_{clean}	1 hr
P_A	$ 2.80 / mol A	$Temp_{ext}$	500 K
P_B	$ 1.70 / mol B	$Temp_i$	300 K
P_D	$ 1.40 / mol D	$Temp_{max}$	480 K
P_{rA}	$ -0.56 / mol A recovered	γ	0.25
P_{rB}	$ -0.36 / mol B recovered		

Additional Nomenclature Introduced in Appendix A.4.

C_A	Concentration of A, mol / l.
C_B	Concentration of B, mol / l.
C_C	Concentration of C, mol / l.
C_D	Concentration of D, mol / l.
C_{tot}	Total amount of product C to be produced, mols.
n_r	Number of reactors available.
P_B	Cost of reactant B, $ / mol.
P_D	Cost of disposing of waste D, $ / mol.
P_{rB}	Cost of disposing of waste or credit for recycle, $ / mol.
R_{drain}	Draining rate, l / hr.
R_{fill}	Filling rate, l / hr.
R_{trans}	Rate of material transfers, l / hr.
$Rate_{tot}$	Average rate of process, mol / hr.
$Rate_{tot\ p}$	Sum of rates for all parallel reactors, mol / hr.
t_{clean}	Time required to clean-out the reactor in preparation for the next batch, hrs.
t_{drain}	Time required to empty reactor, hr.
t_{fill}	Time required to charge reactor, hr.
t_{heat}	Time required for heating up reactants, hr.
$T_{r\,i}$	Cycle time for reactor i, hr.
$Temp_{ext}$	Temperature of heating source, K.
$Temp_i$	Initial reactant feed temperature, K.
x_{Ao}	Initial mole fraction of A.
x_{Bo}	Initial mole fraction of B.
x_D	Mole fraction of D.
$y_{i\,r}$	Existence of unit i at reaction stage.
γ	Proportionality constant in heating time expression.

Appendix A.5: Model Equations for Two Stage Test Problem

The model equations for the two stage test problem described in Section 6.4.3 are summarized here. This two stage problem builds upon the one stage problem described in Appendix A.4. After the batch reaction (see Appendix A.4), the mixture of A, B, C, and D is separated by batch distillation. The relationship among the magnitudes of the relative volatilities is shown below:

$$\alpha_{DB} > \alpha_{CB} > \alpha_{AB} > \alpha_{BB} \tag{A-77}$$

Thus, for the perfect split column model, only C and D are collected overhead. The operation time for columns is given by:

$$t_{c_i} = \frac{(x_D + x_C)\, V_{c_i}\, C}{F_{d_i}} \tag{A-78}$$

The cycle time for column i is:

$$T_{c_i} = 1 + t_{clean_c} + t_{c\ idle\ i} + t_{c_i} + \frac{V_{c_i}}{R_{fill}} + \frac{(x_A + x_B)\, V_{c_i}}{R_{fill}} \tag{A-79}$$

The average rate for the distillation stage is:

$$Rate_{tot_c} = \sum_{i=1}^{n_c} \frac{y_{i_c}\, V_{c_i}\, C\, x_C}{T_{c_i}} \tag{A-80}$$

The limiting rate for the process is then:

$$Rate_{tot} = \min \{Rate_{tot_r}, Rate_{tot_c}\} \tag{A-81}$$

Adding a distillation stage changes the cost expressions for equipment rental, utilities usage, and unit clean-outs. These new cost terms are:

$$\phi_{raw\ materials} = (x_{Ao}\, P_A + x_{Bo}\, P_B + x_B\, P_{rB} + x_D\, P_D)\, \frac{C_{tot}}{x_C} \tag{A-82}$$

$$\phi_{clean} = \sum_{i=1}^{n_r} C_{clr}\, \frac{y_{i_r}\, T_{tot}}{T_{r_i}} + \sum_{i=1}^{n_c} C_{clr}\, \frac{y_{i_c}\, T_{tot}}{T_{c_i}} \tag{A-83}$$

$$\phi_{\text{utilities c}} = P_{uc}\,(x_C + x_D)\,C\,(\text{Ref} + 1)\sum_{i=1}^{n_c}\frac{V_{c_i}\,y_{i_c}}{T_{c_i}}\,T_{tot} \qquad \text{(A-84a)}$$

$$\phi_{\text{utilities r}} = \left(\sum_{i=1}^{n_r}\frac{V_{r_i}}{T_{r_i}}\,y_{i_r}\right)P_{ur}\,C\,(\text{Temp} - \text{Temp}_i)\,T_{tot} \qquad \text{(A-84b)}$$

$$\phi_{\text{util}} = \phi_{\text{utilities}_r} + \phi_{\text{utilities}_c} \qquad \text{(A-85)}$$

$$\phi_{\text{equipment}} = \left[(\sum_{i=1}^{n_r} r_{r_i}\,y_{i_r} + \sum_{i=1}^{n_c} r_{c_i}\,y_{i_c}) + C_s\,(\sum_{i=1}^{n_r} V_{r_i}\,y_{i_r} + \sum_{i=1}^{n_c} V_{c_i}\,y_{i_c})\right]\,T_{tot} \qquad \text{(A-86)}$$

$$\phi = \phi_{\text{raw materials}} + \phi_{\text{waste}} + \phi_{\text{clean-out}} + \phi_{\text{equipment}} + \phi_{\text{utilities}} \qquad \text{(A-87)}$$

The equipments inventory is shown in Table A-6 for this two stage process. Problem parameter values are specified in Table A-7.

TABLE A-6.
EQUIPMENT INVENTORY FOR TWO STAGE PROCESS

REACTORS		
Reactor #	**Size (l)**	**Rental Cost ($/hr)**
1	1000	56.80
2	2000	86.10
3	2500	99.60
4	3500	122.00
5	4000	130.00
COLUMNS		
V_c (l)	**F_d (mol / hr)**	**C_c (mol / l)**
500	1200	50
1000	1800	77
3000	3000	160
5000	4500	300

TABLE A-7
PARAMETER VALUES FOR TWO STAGE PROBLEM

C	5 mol / l	P_u	0.0025
C_{clc}	$ 1221	P_{uc}	$ 1 / mol vaporized
C_{clr}	$ 916 / batch	pre	185,200 l / mol-hr
C_s	0.005 $ / hr-l	R_{trans}	2,000 l / mol
C_{tot}	500,000 mols C	Ref	3
E	12.9 kcal / mol	$t_{clean\ c}$	0.5 hr
P_A	$ 3.20 / mol A	$t_{clean\ r}$	1 hr
P_B	$ 1.70 / mol B	t_{st}	1 hr
P_D	$ 1.40 / mol D	$Temp_{ext}$	500 K
P_{rA}	$ -0.86 / mol A recovered	$Temp_i$	300 K
P_{rB}	$ -0.36 / mol B recovered	γ	0.25

Additional Nomenclature Introduced in Appendix A.5.

$Rate_{tot\ c}$	Sum of rates for all parallel columns, mol / hr.
Ref	Reflux ratio.
Subscript i	Indicates the appropriate value when using unit i.
$t_{clean\ c}$	Time required to clean the column, hr.
t_{idle}	Idle time for column, hr.
t_{st}	Start up time for column, hr.
x_D	Mole fraction of D leaving reactor.
$y_{i\ c}$	Existence of unit i at column stage.
α_{AB}	Relative volatility of A with respect to B.
α_{BB}	Relative volatility of B with respect to B.
α_{CB}	Relative volatility of C with respect to B.
α_{DB}	Relative volatility of D with respect to B.

Appendix A.6: Chapter 7 Data

The reactions involved in the industrial problem described in Chapter 7 are described first.

Reaction I The overall reactions involved in Reaction I are:

$$2R1 + R2 \rightarrow A \tag{A-88}$$

$$R1 + R2 \rightarrow C + G \tag{A-89}$$

The elementary reaction network for Reaction I is shown below:

$$R1 + R2 \rightarrow I1 \qquad \Delta H = 23.0\text{kcal/mol} \tag{A-90}$$

$$R1 + I1 \rightarrow A \qquad \Delta H = 15.0\text{kcal/mol} \tag{A-91}$$

$$I1 \rightarrow C + G \qquad \Delta H = 5.0\text{kcal/mol} \tag{A-92}$$

$$I1 + C \rightarrow I2 \tag{A-93}$$

$$CAT \rightarrow CAT^* \tag{A-94}$$

Detailed kinetic expressions for each species in terms of concentration are (assuming $\Delta V_{rxn} = 0$):

$$\frac{dR_1}{dt} = -k_1 R_1 R_2 \frac{CAT}{k_7 + CAT} - k_2 R_1 I_1 \tag{A-95}$$

$$\frac{dR_2}{dt} = -k_1 R_1 R_2 \frac{CAT}{k_7 + CAT} \tag{A-96}$$

$$\frac{dCAT^*}{dt} = k_6 CAT \tag{A-97}$$

$$\frac{dCAT}{dt} = -k_6 CAT \tag{A-98}$$

$$\frac{dI_1}{dt} = k_1 R_1 R_2 \frac{CAT}{k_7 + CAT} - k_2 R_1 I_1 - k_3 I_1 - k_4 C I_1 + k_5 I_2 \tag{A-99}$$

$$\frac{A}{dt} = k_2 R_1 I_1 \tag{A-100}$$

$$\frac{dC}{dt} = k_3 I_1 - k_4 C I_1 + k_5 I_2 \tag{A-101}$$

$$\frac{dI_2}{dt} = k_4 C I_1 - k_5 I_2 \tag{A-102}$$

All the kinetic rate constants are of the Arrhenius form.

$$k_i = k_{i0} exp^{-E_a/RT} \tag{A-103}$$

Table A-8: Kinetic Data for Reaction I - Case Study Problem

Kinetic constant	Pre-exponential factor	Activation Energy cal/mol-K
k_1	2.70E+11	18700.
k_2	3.63E+06	10900.
k_3	4.40E+14	24700.
k_4	1.29E+05	7700.
k_5	2.64E+13	21800.
k_6	5.00E-01	0.
k_7	7.00E-04	0.

The values for the pre-exponential factors and the activation energies are shown in Table A-8.

Reaction II Reaction II involves converting intermediate C into intermediate E by reaction with R3:

$$\mathrm{C} + \mathrm{R3} \rightarrow \mathrm{E} \qquad (\mathrm{A}-104)$$

The rate expression for Reaction II is

$$\frac{dE}{dt} = kR_3C \qquad (\mathrm{A}-105)$$

where k =1.00 l/mol-h at 75 C.

Reaction III Reaction III converts intermediate E to final product D by reaction with R4.

$$\mathrm{R4} + 2\mathrm{E} \rightarrow \mathrm{D} + 2\mathrm{R3} \qquad (\mathrm{A}-106)$$

for which the kinetic equation is

$$\frac{dE}{dt} = -kE^2 \qquad (\mathrm{A}-107)$$

where $k = 1.79E + 12exp^{-19900/1.987T}$ with T in K.

The second part of Appendix A.6 provides the required physical property data for simulating the case-study. We first present the pure component properties, followed by binary interaction parameters used in the liquid solution model.

Table A-9: Pure Component Properties

Component	Property	Value	Units
A	Boiling Point	532.	K
	Molecular Weight	250.48	
	Critical Temperature	653.58	K
	Critical Pressure	0.16103E+07	N/SQM
	Critical Volume	0.6760	CUM/KMOL
	Critical Compressibility Factor	0.2003	
	Vapor Pressure At TB	0.10132E+06	N/SQM
	At 0.9*TC	0.42209E+06	
	At TC	0.16103E+07	
	Acentric Factor	1.2824	
	Heat of Vap at TB	0.62857E+08	J/KMOL
	Liquid Mol Vol at TB	0.19257	CUM/KMOL
	Solubility Parameter	19281.	(J/CUM)**.5
	Ideal gas CP at 300 K	0.32267E+06	J/KMOL-K
	at 500 K	0.47868E+06	
	at 1000 K	0.69720E+06	
	Molar Volume	0.17045	
E	Boiling Point	416.48	K
	Molecular Weight	222.43	
	Critical Temperature	525.42	K
	Critical Pressure	0.17237E+07	N/SQM
	critical Volume	0.5401	CUM/KMOL
	Critical Compressibility Fcctor	0.2131	
	Vapor Pressure at TB	0.10132E+06	N/SQM
	at 0.9*TC	0.52053E+06	
	at TC	0.17237E+07	
	Acentric Factor	1.0432	
	Heat of Vap at TB	0.45471E+08	J/KMOL
	Liquid Mol Vol at TB	0.17095	CUM/KMOL
	Solubility Parameter	18156.	(J/CUM)**.5
	Ideal gas CP at 300 K	0.27867E+06	J/KMOL-K
	at 500 K	0.40862E+06	
	at 1000 K	0.59384E+06	
	Molar Volume	0.19278	

Table A-9: Pure Component Properties - contd.

I2	Boiling Point	618.27	K
	Molecular Weight	366.72	
	Critical Temperature	757.43	K
	Critical Pressure	0.10718E+07	N/sqm
	Critical Volume	1.2749	CUM/KMOL
	Critical Compressibility factor	0.2170	
	Vapor Pressure at TB	0.10132E+06	N/SQM
	at 0.9*TC	0.33472E+06	
	at TC	0.10718E+07	
	Acentric factor	0.9688	
	Heat of Vap at TB	0.60603E+08	J/KMOL
	Liquid Mol Vol at TB	0.43351	CUM/KMOL
	Solubility Parameter	12946.	(J/CUM)**.5
	Ideal gas CP at 300 K	0.47741E+06	J/KMOL-K
	at 500 K	0.70415E+06	
	at 1000 K	0.10271E+07	
	Molar Volume	0.19934	
D	Boiling Point	752.66	K
	Molecular Weight	398.79	
	Critical Temperature	939.80	K
	Critical Pressure	0.10252E+07	N/SQM
	Critical Volume	1.7592	CUM/KMOL
	Critical Compressibility Factor	0.2308	
	Vapor Pressure at TB	0.10132E+06	N/SQM
	at 0.9*TC	0.36408E+06	
	at TC	0.10252E+07	
	Acentric Factor	0.7469	
	Heat of Vap at TB	0.66080E+08	J/KMOL
	Liquid Mol Vol at TB	0.64668	CUM/KMOL
	Solubility Parameter	11567.	(J/CUM)**.5
	Ideal gas CP at 300 K	0.49038E+06	J/KMOL-K
	at 500 K	0.72412E+06	
	at 1000 K	0.10467E+07	
	Molar Volume	0.22028	

Table A-9: Pure Component Properties - contd.

I1	Boiling Point	373.97	K
	Molecular Weight	176.33	
	Critical Temperature	485.19	K
	Critical Pressure	0.21259E+07	N/SQM
	Critical Volume	0.4169	CUM/KMOL
	Critical Compressibility Factor	0.2197	
	Vapor Pressure at TB	0.10132E+06	N/SQM
	at 0.9*TC	0.68797E+06	
	at TC	0.21259E+07	
	Acentric Factor	0.9293	
	Heat of Vap at TB	0.40288E+08	J/KMOL
	Liquid Mol Vol at TB	0.13591	CUM/KMOL
	Solubility Parameter	18789.	(J/CUM)**.5
	Ideal gas CP at 300 K	0.24064E+06	J/KMOL-K
	at 500 K	0.35527E+06	
	at 1000 K	0.52282E+06	
	Molar Volume	0.20362	
C	Boiling Point	336.57	K
	Molecular Weight	158.24	
	Critical Temperature	494.01	K
	Critical Pressure	0.30423E+07	N/SQM
	Critical Volume	0.5015	CUM/KMOL
	Critical Compressibility Factor	0.3715	
	Vapor Pressure at TB	0.10131E+06	N/SQM
	at 0.9*TC	0.13574E+07	
	at TC	0.30423E+07	
	Acentric Factor	0.3490	
	Heat of Vap at TB	0.29323E+08	J/KMOL
	Liquid Mol Vol at TB	0.13159	CUM/KMOL
	Solubility Parameter	15277.	(J/CUM)**.5
	Ideal gas CP at 300 K	0.20487E+06	J/KMOL-K
	at 500 K	0.33357E+06	
	at 1000 K	0.53086E+06	
	Molar Volume	0.25857	

Table A-9: Pure Component Properties - contd.

R2	Boiling Point	346.00	K
	Molecular Weight	134.33	
	Critical Temperature	512.55	K
	Critical Pressure	0.31036E+07	N/sqm
	Critical Volume	0.3775	CUM/KMOL
	Critical Compressibility factor	0.2749	
	Vapor Pressure at TB	0.10132E+06	N/SQM
	at 0.9*TC	0.14084E+07	
	at TC	0.31036E+07	
	Acentric factor	0.3174	
	Heat of Vap at TB	0.29694E+08	J/KMOL
	Liquid Mol Vol at TB	0.13578	CUM/KMOL
	Solubility Parameter	15305.	(J/CUM)**.5
	Ideal gas CP at 300 K	0.15815E+06	J/KMOL-K
	at 500 K	0.23977E+06	
	at 1000 K	0.35377E+07	
	Molar Volume	0.26156	
R1	Boiling Point	370.00	K
	Molecular Weight	58.08	
	Critical Temperature	545.	K
	Critical Pressure	0.57147E+07	N/SQM
	Critical Compressibility Factor	0.2560	
	Acentric Factor	0.6300	
	Heat of Vap at TB	0.39984E+08	J/KMOL
	Liquid Mol Vol at TB	0.07614	CUM/KMOL
	Solubility Parameter	25679.	(J/CUM)**.5
	Molar Volume	0.25665	

Table A-9: Pure Component Properties - contd.

R3	Boiling Point	337.80	K
	Molecular Weight	32.042	
	Critical Temperature	512.60	K
	Critical Pressure	0.80959E+07	N/sqm
	Critical Compressibility factor	0.2240	
	Acentric factor	0.5590	
	Heat of Vap at TB	0.35278E+08	J/KMOL
	Liquid Mol Vol at TB	0.04350	CUM/KMOL
	Solubility Parameter	29669.	(J/CUM)**.5
	Molar Volume	0.23768	
S	Boiling Point	383.80	K
	Molecular Weight	92.141	
	Critical Temperature	591.70	K
	Critical Pressure	0.41138E+07	N/SQM
	Critical Compressibility Factor	0.2640	
	Acentric Factor	0.2570	
	Heat of Vap at TB	0.333201E+08	J/KMOL
	Liquid Mol Vol at TB	0.11833	CUM/KMOL
	Solubility Parameter	18272.	(J/CUM)**.5
	Molar Volume	0.26498	
R4	Boiling Point	373.20	K
	Molecular Weight	18.015	
	Critical Temperature	647.30	K
	Critical Pressure	0.22048E+07	N/sqm
	Critical Compressibility factor	0.2290	
	Acentric factor	0.3440	
	Heat of Vap at TB	0.40683E+08	J/KMOL
	Liquid Mol Vol at TB	0.01964	CUM/KMOL
	Solubility Parameter	48146.	(J/CUM)**.5
	Molar Volume	0.24317	

Table A-10: Temperature Dependent Properties

Coef.	A	E	I2	D	I1	C
A_1	-20884.	1143.0	-6784.0	-24284.	-657.00	-0.13542E+06
A_2	1400.3	1113.1	1941.7	2095.6	961.38	1522.7
A_3	-0.92240	-0.67220	-1.1714	-1.3704	-0.56020	-1.4824
A_4	0.24E-03	0.15E-03	0.26E-03	0.34E-03	0.12E-03	0.62E-03
A_5	0.	0.	0.	0.	0.	0.
A_6	0.	0.	0.	0.	0.	0.
A_7	280.00	280.00	280.00	280.00	280.00	280.00
A_8	1100.0	1100.0	1100.0	1100.0	1100.0	1100..0
A_9	36029.	36029.	36029.	36029.	36029.	36029
A_{10}	57.226	48.538	88.532	91.050	40.822	25.582
A_{11}	1.5000	1.5000	1.5000	1.5000	1.5000	1.5427
B_1	156.62	128.43	127.31	106.49	115.72	61.623
B_2	-17059.	-11654.	-16097.	-16630.	-9833.5	-5620.8
B_3	0.	0.	0.	0.	0.	0.
B_4	0.	0.	0.	0.	0.	0.
B_5	-18.041	-14.768	-13.991	-11.021	-13.169	-5.7449
B_6	0.93E-17	0.29E-16	0.31E-17	0.71E-18	0.43E-16	0.21E-16
B_7	6.0000	6.0000	6.0000	6.0000	6.0000	6.0000
B_8	532.00	416.48	618.27	752.66	373.97	336.57
B_9	653.58	525.42	757.43	939.80	485.19	494.01

Temperature Dependent Properties

Ideal Gas Heat Capacity (KJ/KMOL-K):

$$\text{CPIG} = A_1 + A_2T + A_3T^2 + A_4^3 + A_5^4 + A_6^5 for A_7 \leq T \leq A_8 \quad \text{(A-108)}$$

$$\text{CPIG} = A_9 + A_{10}T^{A_{11}} \text{ for t} < A_7 \quad \text{(A-109)}$$

Extended Range Antoine Vapor Pressure Equation:

$$\log P_s[kN/m^2] = B_1 + \frac{B_2}{T + B_3} + B_4T + B_5 \log T + B_6T^{B_7} \text{ for } B_8 \leq T \leq B_9 \quad \text{(A-110)}$$

The coefficients A_1 through A_{11} and B_1 through B_9 are presented in Table A-10.

Binary UNIQUAC Parameters

Table A-10: Temperature Dependent Properties - contd.

Coef.	R1	R2	R4	R3	S
A_1	-29644.	-1105.3	33738.	21152.	-24355.
A_2	783.51	314.64	-7.0176	70.924	512.46
A_3	-0.57862	-0.20319	0.27296E-01	0.25870E-01	-0.27654
A_4	0.17853E-03	0.53214E-04	-0.16647E-04	-0.28516E-04	0.49111E-04
A_5	0.	0.	0.	0.	0.
A_6	0.	0.	0.	0.	0.
A_7	280.00	300.00	200.00	300.00	300.00
A_8	1100.0	1272.8	3000.0	1261.8	1665.4
A_9	36029.	33256.	33256.	33256.	33256.
A_{10}	23.961	8.3101	0.18978E-19	0.37813E-01	13.965
A_{11}	1.5000	1.5000	9.2846	2.2014	1.5000
B_1	58.959	100.41	65.154	58.710	71.277
B_2	-5591.9	-8482.7	-6842.9	-6364.7	-6413.3
B_3	0.	0.	0.	0.	0.
B_4	0.0	0.43966E-02	0.27835E-02	-0.23901E-02	0.41663E-02
B_5	-5.3538	-11.435	-6.1364	-4.7344	-7.5054
B_6	0.16714E-16	0.13959E-16	0.33117E-17	0.20888E-16	0.54200E-17
B_7	6.0000	6.0000	6.0000	6.0000	6.0000
B_8	346.00	318.00	319.27	288.67	318.72
B_9	512.55	545.00	647.30	512.60	591.70

The liquid interactions between components i and j were represented by the UNIQUAC liquid solution model in the following form:

$$\tau_{ij} = exp(A2_{ij}/T) \qquad (A-111)$$

The binary interaction parameters for all possible pairs of components are shown in Table A-11.

The third part of Appendix A.6 provides costing information that was used in the case-study.

Equipment Inventory

Unit type	Volume (gal)	Usage cost ($/hr)
Reactors	500	50
	500	50
	750	70
	750	70
	1000	88

Unit type	# trays	Volume (gal)	Vapor (kmol/hr) Rate	Usage cost ($/hr)
Columns	8	750	15	90
	8	750	20	98
	8	1000	20	110
	8	1250	15	125
Combination reactor/column	8	2000	10	175

Intermediate Storage is available in any total amount but only in increments of 250 gallons. Storage costs $ 5 / hr (250 gal increment).

Cost of Raw Materials

Table A-11: Binary UNIQUAC Parameters

Component i	Component j	$A2_{ij}$
A	S	60.52
S	A	-249.7
A	E	25.86
E	A	-31.65
A	R1	74.17
R1	A	-232.4
A	I2	48.59
I2	A	-80.25
A	R3	-167.1
R3	A	-44.31
A	R4	-236.4
R4	A	-192.1
A	D	43.24
D	A	-44.68
A	I1	106.0
I1	A	-127.0
A	C	435.9
C	A	-968.2
S	E	-274.7
E	S	97.94
S	R1	-173.473
R1	S	17.782
S	I2	-350.7
I2	S	97.84
S	R3	-563.46
R3	S	24.3945
S	R4	-1019.
R4	S	-372.1
S	D	-333.3
D	S	106.2
S	I1	-270.4
I1	S	48.65
S	C	-2.727
C	S	-28.47

Table A-11: Binary UNIQUAC Parameters - contd.

Component i	Component j	$A2_{ij}$
E	R1	74.33
R1	E	-253.6
E	I2	19.43
I2	E	-37.62
E	R3	-177.7
R3	E	-44.49
E	R4	218.2
R4	E	-667.9
E	D	21.01
D	E	-13.85
E	I1	73.64
I1	E	-88.72
E	C	452.3
C	E	-1044.
R1	I2	-301.8
I2	R1	-19.52
R1	R3	88.78
R3	R1	-126.5
R1	R4	261.5
R4	R1	-640.6
R1	D	-321.1
D	R1	55.13
R1	I1	-268.6
I1	R1	48.76
R1	C	50.58
C	R1	-496.6
I2	R3	-341.2
R3	I2	-109.3
I2	R4	-521.4
R4	I2	-289.0

Table A-11: Binary UNIQUAC Parameters - contd.

Component i	Component j	$A2_{ij}$
I2	D	-144.3
D	I2	119.0
I2	I1	121.8
I1	I2	-147.9
I2	C	469.3
C	I2	-1137.
R3	R4	165.26
R4	R3	-254.73
R3	D	-81.28
D	R3	-224.7
R3	I1	-87.63
I1	R3	-243.6
R3	C	127.2
C	R3	-868.3
R4	D	-284.1
D	R4	-339.0
R4	I1	-241.4
I1	R4	-347.7
R4	C	-169.2
C	R4	-1579.

Material	Cost ($/unit)
R1	4.11 /kg
R2	8.85/kg
S	1.464 /kg
CAT	35.0 /l slurry
R3	1.23 /kg
R4	0.01 /kg

Cost of Waste Treatment

Wastes	Source	Cost ($/kg)
Solids (CAT, I2, A)	Bottoms Distillation II	13.60
Aqueous (R3, R4, S, E)	Overheads Distillation III	1.70
Organic	Overheads Distillation I	90% S, R3, R1, recoverable at 50% credit on raw material cost

Cost of Utilities ($/batch)

Reaction I heatup	$0.00697(T_{feed,C} - 25)(\text{Final reaction volume}_l)$
Jacket cooling	$4.585E-03(\text{Heating rate}_{kcal/h})(T_{rxn,h})$
Condenser cooling	$0.000655(\text{Condensing rate}_{kcal/h})(T_{rxn,h})$
Reaction I cooldown	$0.01(T_{final,C} - 75)(\text{Final reaction volume}_l)$
Column boilup/condensing	$0.0177(\text{Vapor rate}_{mol/h})(T_{dist,h})$
Vacuum	$0.0011(760 - P_{mmHg})^2(T_{dist,h} + 0.3)$