

**A Multiobjective, Multidisciplinary Design Optimization Methodology  
for the Conceptual Design of Distributed Satellite Systems**

by  
**Cyrus D. Jilla**

B.S., Aerospace Engineering  
University of Virginia, 1996

S.M., Aeronautics & Astronautics  
Massachusetts Institute of Technology, 1999

SUBMITTED TO THE DEPARTMENT OF AERONAUTICS & ASTRONAUTICS  
IN PARTIAL FULFILLMENT OF THE DEGREE OF

**DOCTORATE OF PHILOSOPHY**  
at the  
**MASSACHUSETTS INSTITUTE OF TECHNOLOGY**

May 2002

© Massachusetts Institute of Technology 2002. All rights reserved

Signature of Author .....  
Department of Aeronautics and Astronautics  
May 1, 2002

Certified by .....  
Professor David W. Miller  
Department of Aeronautics and Astronautics  
Thesis Supervisor

Certified by .....  
Professor Daniel E. Hastings  
Department of Aeronautics and Astronautics

Certified by .....  
Charles E. Boppe  
Department of Aeronautics and Astronautics

Certified by .....  
Dr. Raymond J. Sedwick  
Department of Aeronautics and Astronautics

Accepted by .....  
Wallace E. Vander Velde, Professor of Aeronautics and Astronautics  
Chair, Committee on Graduate Students



# **A Multiobjective, Multidisciplinary Design Optimization Methodology for the Conceptual Design of Distributed Satellite Systems**

by  
**Cyrus D. Jilla**

Submitted to the Department of Aeronautics and Astronautics  
on May 1st, 2002, in Partial Fulfillment of the  
Requirements for the Degree of  
Doctorate of Philosophy

## **Abstract**

A multiobjective, multidisciplinary design optimization methodology for mathematically modeling the distributed satellite system (DSS) conceptual design problem as an optimization problem has been developed to advance the state-of-the-art in complex distributed satellite network design. An increasing number of space missions are utilizing DSS architectures in which multiple satellites work in a coordinated fashion to improve system performance, cost, and survivability. The trade space for distributed satellite systems can be enormous – too large to enumerate, analyze, and compare all possible architectures. The seven-step methodology enables an efficient search of the trade space for the best families of architectures, and explores architectures that might not otherwise be considered during the conceptual design phase, the phase of a DSS program in which the majority of lifecycle cost gets locked in.

Four classes of multidisciplinary design optimization (MDO) techniques are investigated – Taguchi, heuristic, gradient, and univariate methods. The heuristic simulated annealing (SA) algorithm found the best DSS architectures with the greatest consistency due to its ability to escape local optima within a nonconvex trade space. Accordingly, this SA algorithm forms the core single objective MDO algorithm in the methodology. The DSS conceptual design problem scope is then broadened by expanding from single objective to multiobjective optimization problems, and two variant multiobjective SA algorithms are developed. The utility in knowing the global Pareto boundary of a DSS trade space is presented, and several methods are explored for approximating the true global Pareto boundary with only a limited knowledge of the full DSS trade space. Finally, methods for improving the performance of the SA algorithm are tested, and it was found that the 2-DOF variant of the SA algorithm is most effective at both single objective and multiobjective searches of a DSS trade space. The versatility of the methodology is demonstrated through its application to the conceptual design of three separate distributed satellite systems – the civil NASA Origins Terrestrial Planet Finder mission, the military TechSat 21 GMTI space-based radar mission, and the commercial broadband satellite communications mission. In each case, the methodology identifies more cost-effective system architectures than those previously considered for the single objective optimization problem, and a Pareto optimal set of architectures for the multiobjective optimization problem. In this manner, the methodology serves as a powerful, versatile systems engineering tool for the conceptual design of distributed satellite systems.

**Thesis Committee Chairman:** David W. Miller, Sc.D.  
Title: Associate Professor of Aeronautics & Astronautics  
Director Space Systems Laboratory

**Thesis Committee Member:** Daniel E. Hastings, Ph.D.  
Title: Professor of Aeronautics & Astronautics and Engineering Systems  
Director Technology and Policy Program  
Associate Director Engineering Systems Division

**Thesis Committee Member:** Charles W. Boppe  
Title: Senior Lecturer  
Department of Aeronautics & Astronautics

**Thesis Committee Member:** Raymond J. Sedwick, Ph.D.  
Title: Research Scientist  
Department of Aeronautics & Astronautics

**Thesis Committee Member:** Andreas S. Schulz, Ph.D.  
Title: Associate Professor of Operations Research  
Sloan School of Management

# TABLE OF CONTENTS

<b>Acknowledgements</b> . . . . .	<b>5</b>
<b>Table of Contents</b> . . . . .	<b>11</b>
<b>List of Figures</b> . . . . .	<b>17</b>
<b>List of Tables</b> . . . . .	<b>25</b>
<b>Acronym List</b> . . . . .	<b>31</b>
<b>Chapter 1. Introduction</b> . . . . .	<b>33</b>
1.1 Motivation . . . . .	33
1.2 Research Context . . . . .	36
1.3 Historical Background . . . . .	39
1.3.1 MDO Literature Review (Aerospace Applications) . . . . .	39
1.3.2 Space Systems Engineering Tools Review . . . . .	43
1.4 Research Objectives and Approach . . . . .	45
1.4.1 Objectives . . . . .	45
1.4.2 Hypothesis . . . . .	47
1.4.3 Approach . . . . .	47
1.4.4 Case Studies . . . . .	48
1.5 Thesis Overview . . . . .	50
<b>Chapter 2. Problem Formulation</b> . . . . .	<b>51</b>
2.1 Systems Engineering Principles . . . . .	51
2.1.1 The Space Systems Design and Development Process . . . . .	51
2.1.2 Systems Engineering Definitions . . . . .	53
2.2 Operations Research Principles . . . . .	55
2.2.1 The Components of an Optimization Problem . . . . .	55
2.2.2 Trade Space Convexity – Local Optima vs. Global Optima . . . . .	56
2.2.3 Optimization Problem Classification . . . . .	59
2.2.4 Solution Techniques . . . . .	60
2.2.5 Single Objective Optimization vs. Multiobjective Optimization . . . . .	61
2.3 DSS Conceptual Design Problem Formulation . . . . .	67
2.3.1 The Standard Formulation . . . . .	67

2.3.2	Sample DSS Conceptual Design Problem Formulation . . . . .	70
2.4	Summary . . . . .	73
<b>Chapter 3.</b>	<b>The MMDOSA Methodology . . . . .</b>	<b>75</b>
3.1	MMDOSA Methodology Overview . . . . .	75
3.2	MMDOSA Step 1 – Create the GINA Model . . . . .	77
3.2.1	Relationship Between GINA and MMDOSA . . . . .	78
3.2.2	GINA Step 1 – Definition of the Mission and CDP Objectives . . . . .	80
3.2.3	GINA Step 2 – Information Network Representation . . . . .	81
3.2.4	GINA Step 3 – Develop System Metrics . . . . .	84
3.2.5	GINA Step 4 – Partition the Conceptual Design Problem . . . . .	96
3.2.6	GINA Step 5 – Develop Simulation Software . . . . .	98
3.2.7	GINA Step 6 – Explore the System Trade Space . . . . .	102
3.2.8	GINA Summary . . . . .	104
3.3	MMDOSA Step 2 – Perform Univariate Studies . . . . .	104
3.4	MMDOSA Step 3 – Take a Random Sample of the Trade Space . . . . .	110
3.4.1	Random Sampling Theory . . . . .	110
3.4.2	TPF Example . . . . .	113
3.4.3	Summary . . . . .	116
3.5	MMDOSA Step 4 – Formulate and Apply MDO Algorithm(s) . . . . .	117
3.5.1	Single Objective Optimization . . . . .	117
3.5.2	Multiobjective Optimization . . . . .	146
3.6	MMDOSA Step 5 – Interpret Results (Sensitivity Analysis) . . . . .	168
3.6.1	Analysis of Variance (ANOVA) . . . . .	169
3.6.2	ANOVA Applied to the TechSat 21 Case Study . . . . .	171
3.6.3	ANOVA Summary . . . . .	173
3.7	MMDOSA Step 6 – Iterate . . . . .	174
3.8	MMDOSA Step 7 – Converge on Best System Architectures . . . . .	178
3.9	MMDOSA Summary – Tying it All Together . . . . .	184
<b>Chapter 4.</b>	<b>Terrestrial Planet Finder Mission Case Study and Results . . . . .</b>	<b>187</b>
4.1	Introduction to the Terrestrial Planet Finder Mission . . . . .	187
4.2	MMDOSA Step 1 – Create the GINA Model . . . . .	188
4.2.1	GINA Step 1 – Definition of the Mission and CDP Objectives . . . . .	189
4.2.2	GINA Step 2 – Information Network Representation . . . . .	190
4.2.3	GINA Step 3 – Develop System Metrics . . . . .	192
4.2.4	GINA Step 4 – Partition the Conceptual Design Problem . . . . .	195

4.2.5	GINA Step 5 – Develop Simulation Software . . . . .	199
4.2.6	GINA Step 6 – Explore the System Trade Space . . . . .	219
4.3	MMDOSA Step 2 – Perform Univariate Studies . . . . .	219
4.4	MMDOSA Step 3 – Take a Random Sample of the Trade Space . . . . .	222
4.5	MMDOSA Step 4 – Formulate and Apply MDO Algorithm(s) . . . . .	225
4.5.1	Approach 1 – Single Objective Optimization . . . . .	226
4.5.2	Approach 2 – Multiobjective Optimization . . . . .	229
4.6	MMDOSA Step 5 – Interpret Results (Sensitivity Analysis) . . . . .	233
4.7	MMDOSA Step 6 – Iterate . . . . .	236
4.8	MMDOSA Step 7 – Converge on Best System Architectures . . . . .	236
4.8.1	Recommended Most Cost-Effective Architectures . . . . .	236
4.8.2	Recommended Pareto Optimal Architectures . . . . .	238
4.9	Conclusions . . . . .	238
 <b>Chapter 5. TechSat 21 Mission Case Study and Results . . . . .</b>		 <b>245</b>
5.1	Introduction to the TechSat 21 Mission . . . . .	246
5.2	MMDOSA Step 1 – Create the GINA Model . . . . .	248
5.2.1	GINA Step 1 – Definition of the Mission and CDP Objectives . . . . .	249
5.2.2	GINA Step 2 – Information Network Representation . . . . .	250
5.2.3	GINA Step 3 – Develop System Metrics . . . . .	252
5.2.4	GINA Step 4 – Partition the Conceptual Design Problem . . . . .	256
5.2.5	GINA Step 5 – Develop Simulation Software . . . . .	259
5.2.6	GINA Step 6 – Explore the System Trade Space . . . . .	302
5.3	MMDOSA Step 2 – Perform Univariate Studies . . . . .	305
5.4	MMDOSA Step 3 – Take a Random Sample of the Trade Space . . . . .	318
5.5	MMDOSA Step 4 – Formulate and Apply MDO Algorithm(s) . . . . .	321
5.5.1	Approach 1 – Single Objective Optimization of The Aerospace Corporation CDC Baseline Design Local Trade Space . . . . .	322
5.5.2	Approach 2 – Single Objective Optimization of the TechSat 21 Global Trade Space . . . . .	325
5.5.3	Approach 3 – Multiobjective Optimization of The Aerospace Corporation CDC Baseline Design Local Trade Space . . . . .	331
5.5.4	Approach 4 – Multiobjective Optimization of the TechSat 21 Global Trade Space . . . . .	335
5.6	MMDOSA Step 5 – Interpret Results (Sensitivity Analysis) . . . . .	342
5.7	MMDOSA Step 6 – Iterate . . . . .	346

5.8	MMDOSA Step 7 – Converge on Best System Architectures . . . . .	346
5.8.1	Recommended Architectures Within The Aerospace Corporation Trade Space . . . . .	347
5.8.2	Recommended Architectures within the Global Trade Space . . . . .	348
5.9	Conclusions . . . . .	350
<b>Chapter 6. Broadband Communications Mission Case Study and Results . . .</b>		<b>353</b>
6.1	Introduction to the Broadband Mission . . . . .	353
6.2	MMDOSA Step 1 – Create the GINA Model . . . . .	354
6.2.1	GINA Step 1 – Definition of the Mission and CDP Objectives . . . . .	355
6.2.2	GINA Step 2 – Information Network Representation . . . . .	356
6.2.3	GINA Step 3 – Develop System Metrics . . . . .	359
6.2.4	GINA Step 4 – Partition the Conceptual Design Problem . . . . .	362
6.2.5	GINA Step 5 – Develop Simulation Software . . . . .	365
6.2.6	GINA Step 6 – Explore the System Trade Space . . . . .	379
6.3	MMDOSA Step 2 – Perform Univariate Studies . . . . .	380
6.4	MMDOSA Step 3 – Take a Random Sample of the Trade Space . . . . .	388
6.5	MMDOSA Step 4 – Formulate and Apply MDO Algorithm(s) . . . . .	390
6.5.1	Approach 1 – Single Objective Optimization . . . . .	391
6.5.2	Approach 2 – Multiobjective Optimization (2 Dimensions) . . . . .	400
6.5.3	Approach 3 – Multiobjective Optimization (4 Dimensions) . . . . .	406
6.6	MMDOSA Step 5 – Interpret Results (Sensitivity Analysis) . . . . .	411
6.7	MMDOSA Step 6 – Iterate . . . . .	412
6.8	MMDOSA Step 7 – Converge on Best System Architectures . . . . .	412
6.8.1	Recommended Most Cost-Effective Architectures . . . . .	412
6.8.2	Recommended 2-Dimensional Pareto Optimal Architectures . . . . .	413
6.8.3	Recommended 4-Dimensional Pareto Optimal Architectures . . . . .	414
6.9	Conclusions . . . . .	415
<b>Chapter 7. Conclusions and Recommendations . . . . .</b>		<b>419</b>
7.1	Thesis Summary . . . . .	419
7.2	Contributions . . . . .	424
7.3	Recommendations for Future Work . . . . .	426
<b>References . . . . .</b>		<b>427</b>
<b>Appendix A. Benefits of Distributed Satellite Systems . . . . .</b>		<b>439</b>



A.1 Overview . . . . .	439
A.2 Potential Benefits of Distribution . . . . .	439
A.2.1 List of the Potential Benefits of Distribution . . . . .	439
A.2.2 Explananation of the Potential Benefits of Distribution . . . . .	440
A.3 Allocation of Potential DSS Benefits to Categories . . . . .	445
A.4 Generic Metrics Matrix . . . . .	446
<b>Appendix B. The Launch Vehicle Selection Optimization Problem . . . . .</b>	<b>449</b>
B.1 Introduction . . . . .	449
B.2 The LST Database . . . . .	452
B.3 Modeling the Launch Vehicle Selection Problem . . . . .	454
B.3.1 Capacitated Network Flow Representation . . . . .	455
B.3.2 Network Design Formulation . . . . .	457
B.3.3 Integer Programming . . . . .	459
B.4 IP Formulation of the Launch Vehicle Selection Problem . . . . .	459
B.4.1 Decision Variables and Objective Function . . . . .	459
B.4.2 Constraints . . . . .	460
B.4.3 IP Model Formulation Summary . . . . .	464
B.5 Integer Program Solution Techniques . . . . .	464
B.6 Case Studies . . . . .	465
B.7 Results . . . . .	467
B.7.1 Test Matrix . . . . .	467
B.7.2 Case Study 1 – Weather Constellation . . . . .	468
B.7.3 Case Study 2 – Telephony Constellation . . . . .	469
B.7.4 Case Study 3 – Broadband Constellation . . . . .	470
B.8 Conclusions . . . . .	471



# LIST OF FIGURES

Figure 1.1	DSS Research Program Overview Slide [Shaw, 1998] . . . . .	38
Figure 1.2	Research Approach . . . . .	48
Figure 2.1	Time Expended vs. Funds/Cost Committed for a Typical Space Project [Casani, 1994] . . . . .	53
Figure 2.2	Example Convex (a) and Concave (b) Functions . . . . .	58
Figure 2.3	Example Nonconvex Function . . . . .	59
Figure 2.4	Five Theoretical Separated Spacecraft Telescope Design Architectures	62
Figure 2.5	The Pareto Boundary in a Two-Dimensional Trade Space . . . . .	66
Figure 2.6	Transformation Between the Classical Optimization Problem and the DSS Conceptual Design Problem . . . . .	67
Figure 3.1	The Multiobjective, Multidisciplinary Design Optimization Systems Architecting Methodology for the Conceptual Design of Distributed Satellite Systems . . . . .	76
Figure 3.2	Relationship Between GINA and MMDOSA . . . . .	79
Figure 3.3	The TPF System Modeled as a Network . . . . .	82
Figure 3.5	Markov Model State Diagram for the Eight-Collector TPF Architecture	87
Figure 3.4	Aggregated Fault Tree for an Eight-Collector TPF Architecture . . . . .	87
Figure 3.6	Markov Model Results for the Eight-Collector TPF Architecture (fcom=5.5x10 <sup>-3</sup> month <sup>-1</sup> , fcol=8.3x10 <sup>-3</sup> month <sup>-1</sup> ) . . . . .	89
Figure 3.7	DSS Lifecycle Costs . . . . .	91
Figure 3.8	Terrestrial Planet Finder Model Component Identification Metrics Matrix [Curtis et al, 1999] . . . . .	98
Figure 3.9	Relationship Between GINA and the Simulation Software Development Process . . . . .	100
Figure 3.10	N2 Diagram for the TPF GINA Model . . . . .	101
Figure 3.11	TPF Cost Per Image vs. Orbit [Curtis et al, 1999] . . . . .	103
Figure 3.12	Wrapping an Optimization Algorithm Around a GINA Model . . . . .	104
Figure 3.13	Sample TechSat 21 Univariate Studies . . . . .	107
Figure 3.14	Example TechSat 21 Design Family . . . . .	108
Figure 3.15	TPF CPI Histograms for the Complete Global Trade Space (a) and the Random Sample of 48 Architectures (b) . . . . .	114

---

Figure 3.16	How the Simulated Annealing Algorithm Escapes Local Optima . . .	125
Figure 3.17	TPF Full-Factorial Trade Space (a) and Trade Space Zoom-In (b) with the True and Taguchi Optimal Solutions . . . . .	133
Figure 3.18	Neighboring TPF Design Vectors when DOF=1 . . . . .	137
Figure 3.19	Trade Space Search as a Function of Simulated Annealing DOF: (a) DOF=1, (b) DOF=3, (c) DOF=5 . . . . .	138
Figure 3.20	Performance of the Simulated Annealing Algorithm in Finding the Minimum CPI TPF System Architecture as a Function of DOF . . . . .	139
Figure 3.21	Simulated Annealing Algorithm Exploration of the TPF Trade Space for (a) DOF=1, (b) DOF=2, (c) DOF=3, and (d) DOF=4 . . . . .	141
Figure 3.22	Local Trade Space (a) and Local Trade Space Zoom-In (b) of The Aerospace Corporation CDC TechSat 21 Baseline Design . . . . .	142
Figure 3.23	Single Objective Simulated Annealing Algorithm Exploration of The Aerospace Corporation CDC TechSat 21 Baseline Design Local Trade Space for DOF=1,2,&3 . . . . .	145
Figure 3.24	Multiobjective Single Solution Simulated Annealing Search of the TPF Trade Space Zoom-In ( $i=48$ iterations) . . . . .	152
Figure 3.25	Multiobjective Multiple Solution Simulated Annealing Search of the TPF Trade Space Zoom-In ( $i=48$ iterations) . . . . .	155
Figure 3.26	TPF Trade Space Global Pareto Boundary . . . . .	159
Figure 3.27	<i>Mean Error</i> Between the Local and Global Pareto Boundary as a Function of the Number of Iterations in Each Algorithm (a) and Approximation of the Global Pareto Boundary by the Multiobjective Multiple Solution Simulated Annealing Algorithm (b) . . . . .	161
Figure 3.28	The Aerospace Corporation TechSat 21 Baseline Design Local Trade Space Pareto Boundary Approximation by 1-DOF (a), 2-DOF (b), and 3-DOF (c) Variants of the Multiobjective Multiple Solution Simulated Annealing Algorithm . . . . .	165
Figure 3.29	Radar Plots Comparing the First Two (a) and All Five (b) Sample Pareto Optimal TechSat 21 Architectures from Table 3.19 . . . . .	167
Figure 3.30	ANOVA Results for The Aerospace Corporation CDC Trade Space with Respect to Lifecycle Cost (a) and Target Probability of Detection (b) . . . . .	172
Figure 3.31	Warm Start of the Single Objective Simulated Annealing Algorithm in the TPF Trade Space . . . . .	176
Figure 3.32	Broadband 2-Dimensional Pareto Optimal Set of System Architectures After One Trial (a) and Five Trials (b) of the Multiobjective Multiple Solution Simulated Annealing Algorithm . . . . .	177

Figure 3.33	TPF Global Trade Space (a) and Zoom-In on the CPI Optimal Design Region (b) . . . . .	180
Figure 3.34	TPF Optimal (i.e. Best) Family of System Architectures . . . . .	181
Figure 3.35	The TechSat 21 Aerospace Corporation CDC Design trade Space (a) and the Pareto Optimal Architectures Located Within the High Performance Region of the Trade Space (b) . . . . .	183
Figure 4.1	A Structurally Connected (a) and Separated Spacecraft (b) TPF Architecture [Beichman et al, 1999] . . . . .	188
Figure 4.2	The TPF System Modeled as a Network . . . . .	191
Figure 4.3	Terrestrial Planet Finder Model Component Identification Metrics Matrix [Curtis et al, 1999] . . . . .	197
Figure 4.4	Alternative View of the TPF Trade Space . . . . .	198
Figure 4.5	TPF GINA Model Simulation Software Block Diagram . . . . .	200
Figure 4.7	Environment Module Inputs and Outputs . . . . .	201
Figure 4.6	N <sup>2</sup> -Diagram for the TPF GINA Model . . . . .	201
Figure 4.8	Aperture Configuration Module Inputs and Outputs . . . . .	203
Figure 4.9	Sample TPF Transmissivity Function [Curtis et al, 1999] . . . . .	204
Figure 4.10	Spacecraft Payload and Bus Module Inputs and Outputs . . . . .	205
Figure 4.11	Dynamics, Control, and Stability Module Inputs and Outputs . . . . .	207
Figure 4.12	Example TPF Finite Element Model . . . . .	208
Figure 4.13	Integrated Dynamics Model Block Diagram [Curtis et al, 1999] . . . . .	209
Figure 4.14	Launch and Operations Module Inputs and Outputs . . . . .	210
Figure 4.15	Systems Analysis Module Inputs and Outputs . . . . .	211
Figure 4.16	Aggregated Fault Tree for an Eight-Collector TPF Architecture . . . . .	214
Figure 4.17	Markov Model State Diagram for the 8-Collector TPF Architecture . . . . .	215
Figure 4.18	Markov Model Results for the Eight-Collector TPF Architecture (fcom=5.5x10 <sup>-3</sup> month <sup>-1</sup> , fcol=8.3x10 <sup>-3</sup> month <sup>-1</sup> ) . . . . .	216
Figure 4.19	Sample TPF GINA Model Benchmark with a TRW Point Design [Curtis et al, 1999] . . . . .	219
Figure 4.20	Example TPF Design Family . . . . .	220
Figure 4.21	Performance Univariate Studies . . . . .	221
Figure 4.22	Cost Per Image Univariate Studies . . . . .	221
Figure 4.23	Lifecycle Cost Univariate Studies . . . . .	222

Figure 4.24	Random Sample Results for Lifecycle Cost (a), Performance (b), and Cost Per Image (c) . . . . .	223
Figure 4.25	TPF Global Trade Space (a) and Trade Space Zoom-In On the Region of Most Cost-Effective Architectures . . . . .	228
Figure 4.26	TPF Trade Space Global Pareto Boundary . . . . .	232
Figure 4.27	Pareto Optimal Architectures Found by the MDO Algorithm . . . . .	233
Figure 4.28	ANOVA Results for the TPF Trade Space . . . . .	234
Figure 4.29	Different Perspectives of the TPF Trade Space as Viewed According to Heliocentric Orbital Radius (a), Collector Connectivity/Geometry (b), Number of Collector Apertures (c), and Aperture Diameter (d) . . . . .	235
Figure 4.30	Most Cost-Effective TPF Design Families . . . . .	237
Figure 4.31	TPF System Trade Space . . . . .	239
Figure 5.1	The TechSat 21 Formation-Flying Cluster Concept [Martin, 2000] . . . . .	247
Figure 5.2	The TechSat 21 System Modeled as a Network . . . . .	251
Figure 5.3	TechSat 21 Model Component Identification Metrics Matrix . . . . .	258
Figure 5.4	Alternative View of the TechSat 21 Trade Space . . . . .	259
Figure 5.5	TechSat 21 GINA Model Simulation Software Block Diagram . . . . .	260
Figure 5.6	$N^2$ Diagram for the TechSat 21 GINA Model . . . . .	261
Figure 5.7	Constellation Module Inputs and Outputs . . . . .	262
Figure 5.8	Sample Polar (a) and Walker (b) TechSat 21 Constellations Generated by the GINA Model . . . . .	263
Figure 5.9	Sample TechSat 21 GINA Model Ground Track (a) and Maximum Revisit Time (b) Plots . . . . .	265
Figure 5.10	Maximum Elevation Angle (a) and Azimuth Angle (b) Probability Density Functions at the Equator for a Sample TechSat 21 Constellation . . . . .	266
Figure 5.11	Radar Module Inputs and Outputs . . . . .	267
Figure 5.12	Radar-Earth Geometry [Cantafio, 1989] . . . . .	270
Figure 5.13	Satellite Orbits in Hill's Frame for an Eight-Satellite TechSat 21 Cluster [Hacker et al, 2000] . . . . .	273
Figure 5.14	Sample Probability of Detection vs. Availability for a TechSat 21 Cluster [Hacker et al, 2000] . . . . .	277
Figure 5.15	Radar Isolation Capability (MDV) (a) and Radar Performance (P(d)) (b) as a Function of Availability Within the System Coverage Area for a Sample TechSat 21 Architecture [Hacker et al, 2000] . . . . .	279
Figure 5.16	Payload Module Inputs and Outputs . . . . .	280

Figure 5.17	TechSat 21 TRAM Module Dimensions of a Single Panel (a) and the Entire Antenna (b) . . . . .	281
Figure 5.18	Spacecraft Bus Module Inputs and Outputs . . . . .	283
Figure 5.19	TechSat 21 Satellite in Stowed (a) and Deployed (b) Configurations [Martin, 2000] . . . . .	284
Figure 5.20	Launch Module (Initial Deployment) Inputs and Outputs . . . . .	286
Figure 5.21	Launch Module (Replenishment) Inputs and Outputs . . . . .	288
Figure 5.22	Aggregated Markov Model State Diagram for an Eight-Satellite TechSat 21 Cluster . . . . .	289
Figure 5.23	Markov Model Results for the Eight-Satellite TechSat 21 Cluster ( $f=0.0002 \text{ month}^{-1}$ , $r=0.33 \text{ month}^{-1}$ ) . . . . .	291
Figure 5.24	Launch Module Outputs for The Aerospace Corporation TechSat 21 Baseline Design . . . . .	293
Figure 5.25	Operations Module Inputs and Outputs . . . . .	294
Figure 5.26	Systems Analysis Module Inputs and Outputs . . . . .	297
Figure 5.27	TechSat 21 GINA Model Graphical User Interface . . . . .	300
Figure 5.28	Mass Comparisons for the TechSat 21 GINA Model Benchmark . . . . .	301
Figure 5.29	Cost Comparisons for the TechSat 21 GINA Model Benchmark . . . . .	301
Figure 5.30	Aerospace Corporation CDC Design Trade Space (a) and Trade Space Zoom-In (b) . . . . .	304
Figure 5.31	Example TechSat 21 Design Family . . . . .	306
Figure 5.32	Lifecycle Cost TechSat 21 Univariate Studies . . . . .	308
Figure 5.33	Probability of Detection TechSat 21 Univariate Studies . . . . .	309
Figure 5.34	Minimum Detectable Velocity TechSat 21 Univariate Studies . . . . .	310
Figure 5.35	Maximum Revisit Time TechSat 21 Univariate Studies . . . . .	311
Figure 5.36	Maximum Cluster Baseline TechSat 21 Univariate Studies . . . . .	312
Figure 5.37	Average SNR TechSat 21 Univariate Studies . . . . .	313
Figure 5.38	Number of Satellites TechSat 21 Univariate Studies . . . . .	314
Figure 5.39	Single Shift Operations Crew Size TechSat 21 Univariate Studies . . . . .	315
Figure 5.40	Software TechSat 21 Univariate Studies . . . . .	316
Figure 5.41	Spacecraft Mass TechSat 21 Univariate Studies . . . . .	317
Figure 5.42	Random Sample Results for Lifecycle Cost (a), Target Probability of Detection (b), and Maximum Revisit Time (c) . . . . .	319

---

Figure 5.43	Local Trade Space (a) and Local Trade Space Zoom-In (b) of The Aerospace Corporation CDC TechSat 21 Baseline Design . . . . .	326
Figure 5.44	The Aerospace Corporation CDC Baseline Design Local Trade Space Pareto Front (a) and Pareto Front Zoom-In (b) . . . . .	336
Figure 5.45	Results from the Application of the Multiobjective Multiple Solution Simulated Annealing Algorithm (DOF=2) on the TechSat 21 Global Trade Space (a) with the AFRL Requirements Superimposed (b) . . . . .	341
Figure 5.46	ANOVA Results for The Aerospace Corporation CDC TechSat 21 Design Trade Space with Respect to Lifecycle Cost (a) and Target Probability of Detection (b) . . . . .	343
Figure 5.47	Different Views of The Aerospace Corporation CDC TechSat 21 Design Trade Space . . . . .	345
Figure 6.1	The Broadband System Modeled as a Network . . . . .	357
Figure 6.2	Broadband Model Component Identification Metrics Matrix . . . . .	363
Figure 6.3	Alternative View of the Broadband Trade Space . . . . .	365
Figure 6.4	Broadband GINA Model Simulation Software Block Diagram . . . . .	366
Figure 6.5	Orbital Dynamics Module Inputs and Outputs . . . . .	367
Figure 6.7	Estimating the Potential Number of Satellite Subscribers . . . . .	368
Figure 6.6	Market Analysis Module Inputs and Outputs . . . . .	368
Figure 6.8	Market Distribution Estimation [Kashitani, 2002] . . . . .	369
Figure 6.9	Link Budget Module Inputs and Outputs . . . . .	370
Figure 6.10	Payload & Bus Module Inputs and Outputs . . . . .	373
Figure 6.11	Launch and Operations Module Inputs and Outputs . . . . .	375
Figure 6.12	Systems Analysis Module Inputs and Outputs . . . . .	376
Figure 6.13	System Performance: Matching Network Capacity to Market Demand [Kashitani, 2002] . . . . .	377
Figure 6.14	Example Broadband Design Family . . . . .	381
Figure 6.15	Lifecycle Cost Univariate Studies . . . . .	382
Figure 6.16	Total Number of Subscriber Years (i.e. System Performance) Univariate Studies . . . . .	383
Figure 6.17	Cost Per T1-Minute Univariate Studies . . . . .	384
Figure 6.18	Mean Coverage Probability Over All Populated Areas (i.e. Entire Potential Market) Univariate Studies . . . . .	385
Figure 6.19	Mean Coverage Probability Over Served Populated Areas (i.e. Availability) Univariate Studies . . . . .	386



---

Figure 6.20	Satellite Mass Univariate Studies . . . . .	387
Figure 6.21	Random Sample Results for Lifecycle Cost (a), System Performance (b), Cost Per T1-Minute (c), Global Availability (d), Served Customers Avail- ability (e), and Satellite Mass (e) . . . . .	389
Figure 6.22	Broadband Trade Space Pareto Boundary . . . . .	404
Figure 6.23	Pareto Optimal Architectures Found by the MDO Algorithm . . . . .	404
Figure 6.24	ANOVA Results for the Broadband Trade Space . . . . .	411
Figure 6.25	Broadband Satellite System Trade Space . . . . .	416
Figure B.1	Trades in the Launch Vehicle (LV) Selection Problem . . . . .	451
Figure B.2	Capacitated Network Flow Model . . . . .	455
Figure B.3	Network Design Model . . . . .	457
Figure B.4	Total Launch Costs for Different Objective Functions in the Three Case Studies . . . . .	468



# LIST OF TABLES

TABLE 1.1	Interpretation of the Terms "Good Solutions" and "Best Solutions" as Used in This Thesis . . . . .	36
TABLE 1.2	Survey of Systems Engineering Tools for Space Systems Conceptual Design [Bearden, 1998; Mosher, 2000] . . . . .	44
TABLE 1.3	MMDOSA Case Studies . . . . .	49
TABLE 2.1	Optimization Problem Taxonomy . . . . .	60
TABLE 2.2	Key Parameters of Five Separated Spacecraft Telescope Architectures	62
TABLE 2.3	Solutions to the Separated Spacecraft Telescope Design Problem . . .	63
TABLE 2.4	The TechSat 21 Design Vector . . . . .	70
TABLE 2.5	Number of "Good" and "Best" Solutions in the Three Case Studies (Single Objective) . . . . .	72
TABLE 3.1	Integrity Requirements for the Different Detection Modes of TPF . .	83
TABLE 3.2	The GINA Capability Quality of Service Metrics For TPF . . . . .	84
TABLE 3.3	Air Force USCM CERs [Larson & Wertz, 1992] . . . . .	92
TABLE 3.4	The Aerospace Corporation SSCM Version 7.4 CERs [Wertz & Larson, 1996] . . . . .	93
TABLE 3.5	The Aerospace Corporation SSCM Version 8.0 CERs [Wertz & Larson, 1996] . . . . .	93
TABLE 3.6	Learning Curve Slopes In the Space Industry . . . . .	94
TABLE 3.7	The Cost Per Function Metric for Different DSS Missions [Shaw et al, 2001] . . . . .	96
TABLE 3.8	The TPF Design Vector . . . . .	97
TABLE 3.9	Baseline TechSat 21 Design Vector . . . . .	106
TABLE 3.9	Baseline TechSat 21 Design Vector . . . . .	108
TABLE 3.10	TPF Population and Random Sample CPI Statistics . . . . .	115
TABLE 3.11	The TPF Design Vector . . . . .	119
TABLE 3.12	Orthogonal Taguchi Analysis Test Matrix for the TPF Case Study . .	123
TABLE 3.13	The Simulated Annealing Algorithm . . . . .	125
TABLE 3.15	Optimized Configuration From Taguchi Analysis . . . . .	130
TABLE 3.14	Taguchi Analysis Results . . . . .	130

TABLE 3.16	Simulated Annealing, Pseudogradient Search, and Univariate Search Algorithm Results . . . . .	132
TABLE 3.17	Squared Error (SE) and Mean Square Error (MSE) of the Single Objective Simulated Annealing Algorithm as a Function of DOF . . . . .	143
TABLE 3.18	ME Between the Approximated and True Pareto Boundary for Each Multi-objective Algorithm Variant . . . . .	164
TABLE 3.19	Five Sample Pareto Optimal TechSat 21 System Architectures . . . . .	166
TABLE 3.20	ANOVA Terminology . . . . .	169
TABLE 3.21	Data From the TPF Warm Starting Example in Figure 3.31 . . . . .	176
TABLE 3.22	Number of Good and Best Solutions in the Three Case Studies . . . . .	179
TABLE 3.23	Pareto Optimal TechSat 21 Architectures from Figure 3.35b . . . . .	184
TABLE 4.1	The GINA Capability Quality of Service Metrics for the TPF GMTI Mission . . . . .	192
TABLE 4.2	The TPF Design Vector . . . . .	196
TABLE 4.3	Baseline TPF Design Vector . . . . .	220
TABLE 4.5	TPF 95% Confidence Intervals for the Global Trade Space Mean . . . . .	224
TABLE 4.4	Statistics from a Random Sample of the TPF Global Trade Space . . . . .	224
TABLE 4.6	MDO Test Matrix for TPF Case Study . . . . .	225
TABLE 4.7	Simulated Annealing Cooling Schedule . . . . .	227
TABLE 4.8	Most Cost-Effective TPF Architectures Found by the Simulated Annealing Algorithm . . . . .	229
TABLE 4.9	Simulated Annealing Cooling Schedule . . . . .	231
TABLE 4.10	Recommended Families of Architectures for Phase B . . . . .	237
TABLE 4.11	Best Pareto Optimal TPF Architectures Ordered as a Function of Program Budget . . . . .	238
TABLE 4.12	Comparison of NASA JPL Point Design with Designs Found by the MMDOSA Methodology . . . . .	240
TABLE 4.13	Comparison of Ball Aerospace Point Design with Designs Found by the MMDOSA Methodology . . . . .	241
TABLE 4.15	Comparison of TRW SSI Point Design with Designs Found by the MMDOSA Methodology . . . . .	242
TABLE 4.14	Comparison of TRW SCI Point Design with Designs Found by the MMDOSA Methodology . . . . .	242
TABLE 4.16	Comparison of Lockheed Point Design with Designs Found by the MMDOSA Methodology . . . . .	243

TABLE 5.1	Space-Based Radar GMTI Concept Comparison [Martin, 2000]	248
TABLE 5.2	The GINA Capability Quality of Service Metrics for the TechSat 21 GMTI Mission	252
TABLE 5.3	The TechSat 21 Design Vector	257
TABLE 5.4	Polar and Walker Constellation Orbital Elements for the TechSat 21 GINA Model	263
TABLE 5.5	Launch Vehicle Cost and Payload Fairing Dimensions Matrix [Isakowitz, 1995]	286
TABLE 5.6	Launch Vehicle Performance Matrix (Polar Orbits, inclination~90°) [Isakowitz, 1995]	287
TABLE 5.7	Aerospace Corporation CDC Point Design Inputs for TechSat 21 GINA Model Benchmark	300
TABLE 5.8	Aerospace Corp. CDC Point Design Inputs for TechSat 21 Local Trade Space Exploration	303
TABLE 5.9	Baseline TechSat 21 Design Vector	306
TABLE 5.11	95% CI's for the TechSat 21 Global Trade Space Mean	320
TABLE 5.10	Random Sample Statistics of the TechSat 21 Global Trade Space	320
TABLE 5.12	MDO Test Matrix for the TechSat 21 Case Study	321
TABLE 5.13	Optimization Approach 1 Design Vector	324
TABLE 5.14	Simulated Annealing Cooling Schedule	324
TABLE 5.15	Minimum CPF Architectures Within the Local Trade Space	327
TABLE 5.17	Simulated Annealing Cooling Schedule	329
TABLE 5.16	Optimization Approach 2 Design Vector	329
TABLE 5.18	Best TechSat 21 Architectures Found within the Global Trade Space	330
TABLE 5.19	Optimization Approach 3 Design Vector	333
TABLE 5.21	Simulated Annealing Cooling Schedule	334
TABLE 5.20	Objective Function Statistics From Complete Enumeration of The Aerospace Corporation CDC Baseline Design Local Trade Space	334
TABLE 5.22	Pareto Optimal TechSat 21 Architectures from Figure 5.44b	337
TABLE 5.23	Optimization Approach 4 Design Vector	338
TABLE 5.24	$\delta_n$ Values for the Multiobjective Optimization of the TechSat 21 Global Trade Space	339
TABLE 5.25	Simulated Annealing Cooling Schedule	340

TABLE 5.26	Best TechSat 21 Architectures Found within the Global Trade Space After Warm Start . . . . .	346
TABLE 5.27	Best Architectures Within the TechSat 21 Aerospace Corporation Baseline Design Trade Space . . . . .	347
TABLE 5.28	Recommended Family of Architectures for Phase B . . . . .	348
TABLE 5.29	Recommended Family of Architectures for Phase B . . . . .	349
TABLE 5.30	Best Multiobjective Architectures Found Within the TechSat 21 Global Trade Space . . . . .	350
TABLE 5.31	Comparison of Existing Designs vs. Design Found by the MMDOSA Methodology . . . . .	351
TABLE 6.1	The GINA Capability Quality of Service Metrics for the Broadband Mission . . . . .	358
TABLE 6.2	The Broadband Design Vector . . . . .	363
TABLE 6.3	Potential Satellite Broadband Subscriber Base [Kashitani, 2002] . . . . .	369
TABLE 6.4	Link Budget Parameters [Kashitani, 2002] . . . . .	371
TABLE 6.5	Launch Vehicle Parameters [Kashitani, 2002] . . . . .	376
TABLE 6.6	Baseline Broadband Design Vector . . . . .	380
TABLE 6.7	Statistics from a Random Sample of the Broadband Trade Space . . . . .	388
TABLE 6.8	Broadband 95% CI's for the Global Trade Space Mean . . . . .	390
TABLE 6.9	MDO Test Matrix for the Broadband Case Study . . . . .	392
TABLE 6.10	Simulated Annealing Cooling Schedule . . . . .	398
TABLE 6.11	Best Business Case Broadband Architectures Found by the MDO Algorithm . . . . .	399
TABLE 6.12	Simulated Annealing Cooling Schedule . . . . .	403
TABLE 6.13	$\delta_n$ Values for the Computation of $\Delta$ . . . . .	408
TABLE 6.14	Simulated Annealing Cooling Schedule . . . . .	409
TABLE 6.15	Four-Dimensional Pareto Optimal Set . . . . .	410
TABLE 6.16	Recommended Family of Architectures for Phase B . . . . .	413
TABLE 6.17	Recommended Broadband Satellite Architectures as a Function of Financial Investment . . . . .	414
TABLE 6.18	Recommended Four-Dimensional Pareto Optimal Broadband Satellite Architectures . . . . .	415
TABLE 6.19	Comparison of Boeing Point Design with Designs Found by the MMDOSA Methodology . . . . .	417

---

TABLE 6.20	Comparison of HughesNET Point Design with Designs Found by the MMDOSA Methodology . . . . .	418
TABLE A.1	Generic Metrics Matrix . . . . .	447
TABLE B.1	Typical Database Entry . . . . .	453
TABLE B.2	Different Objective Function Strategies . . . . .	460
TABLE B.3	IP Formulation of the Launch Vehicle Selection Problem . . . . .	464
TABLE B.4	Characteristics of the Three Case Studies . . . . .	466
TABLE B.5	Case Study 1 Results – 9 Satellite Weather Constellation . . . . .	469
TABLE B.6	Case Study 2 Results – 72 Satellite Telephony Constellation . . . . .	470
TABLE B.7	Case Study 3 Results – 288 Satellite Broadband Constellation . . . . .	471





# ACRONYM LIST

ADCS	Attitude Determination and Control System
AFOSR	Air Force Office of Scientific Research
AFRL	Air Force Research Laboratory
AMTI	Air Moving Target Indication
ANOVA	Analysis of Variance
ASR	Area Search Rate
BER	Bit Error Rate
CDC	Conceptual Design Center
CDP	Conceptual Design Phase
CEM	Concurrent Engineering Methodology
CER	Cost Estimation Relationship
COBRA	Complexity Based Risk Assessment
COTS	Commercial Off the Shelf
CPF	Cost Per Function
CPI	Cost Per Image
DPCA	Displaced Phased Center Antenna
DSL	Digital Subscriber Line
DSS	Distributed Satellite System
ESSAM	Early Small Satellite System Analysis Method
FAR	False Alarm Rate
FCC	Federal Communications Commission
GEO	Geosynchronous Orbit
GINA	Generalized Information Network Analysis Methodology
GMTI	Ground Moving Target Indication
GPS	Global Positioning System
GUI	Graphical User Interface
ICE	Integrated Concurrent Engineering
IFSAR	Interferometric Synthetic Aperture Radar
INCOSE	International Council on Systems Engineering
IP	Integer Program
JPL	Jet Propulsion Laboratory
JSTARS	Joint Surveillance and Target Attack Radar System
KESLOC	Thousands of Estimated Software Lines of Code
LCC	Lifecycle Cost
LEO	Low Earth Orbit
LP	Linear Program
MAUT	Multi-Attribute Utility Theory
MDO	Multidisciplinary Design Optimization
MDV	Minimum Detection Velocity
MEO	Medium Earth Orbit

MERIT	Methodology for Evaluating, Ranking, and Inserting Technology
MIDAS	Multidisciplinary Integrated Design Assistant for Spacecraft
MIT	Massachusetts Institute of Technology
MIP	Mixed Integer Program
MMDOSA	Multiobjective, Multidisciplinary Design Optimization Systems Architect- ing Methodology
NASA	National Aeronautics and Space Administration
NPV	Net Present Value
O-D	Origin-Destination
OPD	Optical Pathlength Difference
PC	Personal Computer
PDC	Product Design Center
PRF	Pulse Repetition Frequency
PTM	Project Trades Model
PVP	Present Value of Profit
RCS	Radar Cross-Section
RTG	Radioisotope Thermoelectric Generator
SAR	Synthetic Aperture Radar
SCI	Structurally Connected Interferometer
SCOUT	Spacecraft Concept Optimization Utility Tool
SE&A	Systems Engineering and Architecting
SMAD	Space Mission Analysis and Design
SNR	Signal-to-Noise Ratio
SPASAT	Space System Analyst's Toolkit
SSCM	Small Satellite Cost Model
SSI	Separated Spacecraft Interferometer
SSL	Space Systems Laboratory
STAP	Space-Time Adaptive Processing
TechSat 21	Technology Satellite for the 21st Century
TFU	Theoretical First Unit
TPF	Terrestrial Planet Finder
TRAM	Transmit/Receive Antenna Module
USCM	Unmanned Spacecraft Cost Model

# Chapter 1

## INTRODUCTION

### 1.1 Motivation

Optimization is defined as the process of achieving the most favorable system condition on the basis of a metric or set of metrics [Merriam-Webster, 1998]. Within the past fifty years, different optimization techniques have been applied to numerous complex problems, ranging from the design of airline flight networks that maximize revenues [Subramanian et al, 1999] under scheduling constraints [Mathaisel, 1997] to the allocation of assets in financial portfolios [Stettner, 1999] under capital, regulatory, and risk constraints. The research presented in this thesis explores the potential of, and develops a framework for, the application of multidisciplinary design optimization (MDO) techniques to the conceptual design of distributed satellite systems (DSS).

As illustrated by the following two quotations, the conceptual design of space systems currently tends to be unstructured, with designers often pursuing a single concept or modifying an existing idea rather than generating new alternatives.

Currently space [systems] are "optimized" manually through a tool assisted evaluation of alternatives and consensus choices about design options and trades. With this approach there is no guarantee that a *systems level focus* will be taken, and the resulting design is usually a collection of high-performance subsystem implementations that when integrated are not a highly efficient system implementation. – [Mosher, 1996]

...the conceptual space [systems] design process is very *unstructured*....design researchers have found that actual design does not follow [an organized] process....designers often pursue a *single design concept*, patching and repairing their original idea rather than generating new alternatives. Conceptual space [systems] design also suffers from this single design concept fixation....these methods [of conceptual space systems design] explore a *limited number of options* with three to four being the limit due to schedule and cost constraints....[current] approaches tend to settle on a single point design very quickly. – [Mosher, 1998]

With the traditional aerospace point design approach, there is no guarantee that a system-level focus will be taken, and often the final design architecture chosen achieves only feasibility instead of optimality [Mosher, 1996]. System-level trades are often delayed until after a point design has been selected because of the perceived time and effort required to conduct a credible analysis [Riddle, 1998]. By not properly exploring the system trade space and converging upon an optimal or even efficient solution during the conceptual design phase, the lifecycle cost of the system can greatly increase as modifications are required to properly integrate and operate the system during the latter stages of the design process, when changes become much more expensive to implement [Shishko & Chamberlain, 1995].

The pitfalls in not following a structured process during conceptual design holds especially true for distributed satellite systems, which tend to be among the most complex and expensive space systems. A distributed satellite system is defined as a system of multiple satellites designed to work together in a coordinated fashion to perform a mission [Shaw et al, 2000]. Examples include the global positioning system (GPS) for navigation, the recently deployed low Earth orbit global mobile communications constellations, and proposed separated spacecraft interferometers for astronomy. The advantages of distributed systems over traditional single satellite deployments – including improvements in performance, cost, and survivability – have led to an increase in the number of civilian, military, and commercial space missions that are considering distributed approaches. The benefits of distributed satellite systems are discussed in greater detail in Appendix A.

---

Distributed satellite systems are among the most challenging systems to design as a large number of highly coupled variables are involved. Take the example of a formation-flying separated spacecraft interferometer designed to image extra-solar planets. Both the total number of spacecraft in the array and the orbit of the interferometer drive the selection of the launch vehicle, which can be a dominant contributor to the system cost. The total number of spacecraft in the array also directly determines the operations complexity (i.e. nonlinear effect on operations cost) and indirectly determines the imaging rate (i.e. scales with the total collecting area). Likewise, the orbit also affects the imaging rate by determining the amount of local zodiacal dust (i.e. imaging noise source) the interferometer must peer through, and so forth [Beichman et al, 1999]. As in all distributed satellite systems, countless trades exist between system performance, system cost, and each of the design parameters, both individually and in combination with other design parameters. Typically, only a handful of point designs are derived from existing, previous generation designs. As a result, the final design is probably inefficient, leaving room for significant improvements in performance and reductions in lifecycle cost. Thus, a method is needed to enable a greater search of the trade space and explore design options that might not otherwise be considered during the conceptual design phase.

Optimization is one such method. In its pure definition, optimization refers to finding the absolute best solution to a problem. This is not the definition that will be used here, however. Rather, the engineering interpretation of optimization, as outlined in Table 1.1, will be referred to as the process of finding good solutions with the intention of finding the best solutions to the conceptual design problem. Because DSS design problems tend to be combinatorial in nature with discrete variables having nonlinear relationships, classical optimization techniques that require continuously differentiable convex functions, like the simplex method, cannot be used. Rather, multidisciplinary design optimization algorithms that can handle discrete variables in nonlinear problems with multiple criteria objective functions and constraints are required. If developed and applied thoughtfully, MDO will allow systems engineers to systematically explore the vast DSS trade space in an intelligent manner and consider many more architectures during the conceptual design phase

before converging on the final design. The research presented in this thesis develops and applies just such a methodology – the multiobjective, multidisciplinary design optimization systems architecting methodology (MMDOSA) for the conceptual design of distributed satellite systems. MMDOSA will provide space systems engineers with a new supporting tool for the conceptual design of distributed satellite systems.

**TABLE 1.1** Interpretation of the Terms "Good Solutions" and "Best Solutions" as Used in This Thesis

<b>Term</b>	<b>Perspective</b>	<b>Definition</b>
Good Solution*	Engineering	Top 0.1%-10% of Solutions in Trade Space
Best Solution*	Engineering	Top 0.01%-1% of Solutions in Trade Space
Best Solution*	Operations Research	Single Best Solution in the Entire Trade Space

\*For a single objective optimization problem.

The goal in applying MDO techniques to the design of distributed satellite systems is not to fully automate the design process and remove humans from the design loop, but rather to facilitate the conceptual design process, considered by many mission leaders to be the most important stage of design, often defining the line between success and failure [Matthews, 1998]. MDO may be able to achieve this by enabling a greater, more efficient exploration of the system trade space to find robust and perhaps even counterintuitive design architectures for further analysis that might not otherwise be considered. Another historical criticism of MDO is that it only works well for small problems with a single objective function. However, the ability of the MMDOSA methodology to handle both single and multiple objective conceptual design problems with over 100 variables and constraints will be demonstrated in three separate case studies in Chapters 4-6 of this thesis.

## 1.2 Research Context

All of the research presented in this thesis flows from a collaborative research program – named the Distributed Satellite Systems program – between the MIT Space Systems Laboratory (SSL) and the Air Force Research Laboratory (AFRL) at Kirtland Air Force Base

in Albuquerque, New Mexico. This research program originated in the Fall of 1996 when the "faster, better, cheaper" mantra had become popular within NASA. At the same time, many people in the defense and intelligence space communities were drawing an analogy between the historical evolution of the computer industry and the potential future evolution of the space industry. Just as the computer industry evolved from single, large and expensive mainframe systems forty years ago to the networked clusters of workstations of today that work collaboratively to solve the same computational problems more cost-effectively, the military space community predicted a transition in space systems from single, large and expensive satellite programs – such as Milstar and the Hubble Space Telescope – to networks of smaller satellites that would work collaboratively to execute the same missions more cost-effectively. While this analogy sounded good from a qualitative perspective, very little quantitative work existed in the technical literature to support the claims made by this analogy concerning improving the cost-effectiveness of space systems. This became the overarching task of the DSS program – to determine quantitatively whether or not this analogy between the historical evolution of computer systems and the potential future evolution of space systems would hold true. Specific questions to be answered by the DSS program included:

- *When* (i.e. for what defense and intelligence missions) does this analogy hold true?
- If this analogy does hold true, *why* does it hold true? (i.e. Are there manufacturing economies of scale that result from producing a greater quantity of smaller satellites? Are there nonlinear improvements in system performance that result from distributed architectures?)
- When this analogy does hold true, *to what extent* does it hold true? (i.e. Should the mission transition from one to five, one to fifty, or one to five hundred satellites?)

Figure 1.1 presents the MIT-AFRL DSS research program overview slide. Column one of the slide lists some of the reasons commonly given in favor of DSS architectures. Appendix A provides a more detailed discussion on the benefits of DSS architectures for space missions. The second column in the slide illustrates the original set of military missions focused on in the DSS program. Chapter 5 details one of these original military mis-

sions – the TechSat 21 mission. Since its conception, the DSS program has branched out to investigate civil (Chapter 4) and commercial (Chapter 6) space missions as well, both of which will be covered in this thesis. Column three in the slide lists some of the key components of the systems engineering methodology, named GINA, that was developed by the MIT Space Systems Laboratory to quantitatively analyze distributed satellite systems [Shaw et al, 2001]. As explained in Chapter 3, the execution of the GINA methodology to create a model of the space system to be architected is the first step of the seven-step MMDOSA methodology developed in this thesis. As listed at the bottom of the slide, the final products of this research effort are formal systems engineering tools that can be used to design future distributed satellite systems. The multiobjective, multidisciplinary design optimization systems architecting methodology for the conceptual design of distributed satellite systems that is developed and presented in this thesis is one such formal tool that has resulted from the DSS research program.

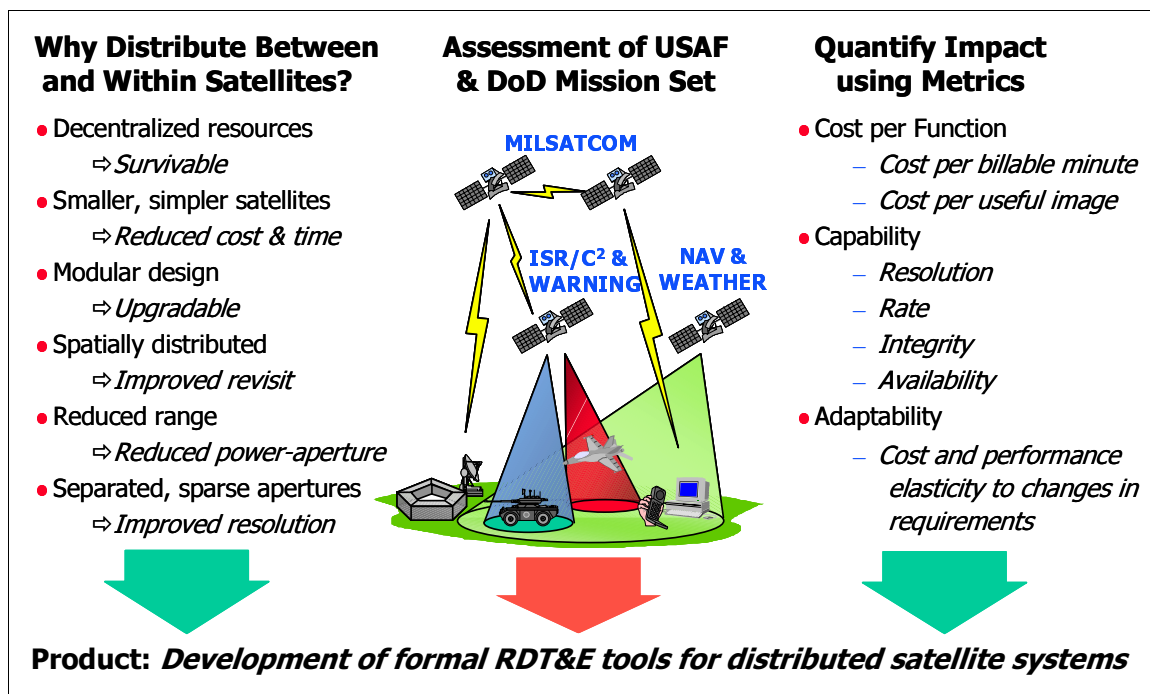


Figure 1.1 DSS Research Program Overview Slide [Shaw, 1998]



## 1.3 Historical Background

This research represents the merging of two distinct fields – multidisciplinary design optimization and space systems engineering. The following two sections detail the literature review on previous attempts to apply MDO to the design of aerospace systems and the existing systems engineering tools for space systems design.

### 1.3.1 MDO Literature Review (Aerospace Applications)

The first formal applications of optimization within the aerospace field occurred within specific specialties. For example, optimization codes are commonly used today by computational fluid dynamicists [Huddleston, 1999; Massardo & Satta, 1989; Sobieczky, 1990] to minimize drag over an aircraft body or component and by structural engineers [Hansen & Vanderplaats, 1990; Hopkins, 1990; Hsu & Adeli, 1991] to design a truss that will safely meet all loading requirements with the minimum possible mass. Another aerospace field that incorporates the principles of optimization is the field of orbital dynamics. For example, the two-impulse Hohmann transfer ellipse is the minimum energy transfer trajectory between two orbits [Hale, 1994]. Similarly, Walker-Delta constellations represent the minimum number of satellites required for single, double, etc. global coverage of the Earth [Walker, 1971].

Thus, the original applications of optimization within the aerospace field entailed optimizing individual components or subsystems – such as the aerodynamic shape, truss structure, and orbital trajectory – and then integrating these subsystems together. However, it has been found that optimized subsystems don't always fit together to produce the best integrated system [Mosher, 1996]. The next logical step in the evolution of the application of optimization to aerospace was to apply MDO principles with the goal of optimizing the *entire* system.

Over the past 15 years, multidisciplinary design optimization has become a rapidly growing field with applications to a wide variety of engineering problems. Within the specific

field of aerospace engineering, most MDO efforts have focused on the design of structures, aircraft, and launch systems. Very little work has considered the application of MDO techniques to distributed satellite systems. The remainder of this section documents the technical literature covering the application of MDO within the field of space systems design.

Within aerospace engineering, MDO was first applied to the design of aircraft and has now evolved from existing merely as an academic exercise to being used in the design of production aircraft [Sobieszczanski-Sobieski, 1990; Kroo et al, 1994; Bos, 1996]. The bulk of MDO work on space applications, which to date has been more academic in nature, has looked at the design of launch vehicles. Researchers in Germany have used a genetic algorithm in combination with a sensitivity penalty function to design a two-stage-to-orbit (TSTO) launch vehicle [Engler et al, 1998]. Another German researcher has investigated applying an MDO approach to select the number of engines, nozzle expansion ratio, mass flow rate, chamber pressure, propellant type, and mixture ratio to maximize the payload capacity or minimize the program cost of a conventional single-stage-to-orbit space transportation system [Kesselman, 1998]. American researchers have been pursuing an alternative MDO approach, known as collaborative optimization, for launch vehicle design [Braun et al, 1995; Braun et al, 1996; Rowell et al, 1996]. Most recently, researchers at the Aerospace Corporation have begun integrating separate models of launch vehicle performance, launch vehicle manufacturing, launch vehicle configurations, facility costs, payload deployment, and launch vehicle cost with the goal of eventually optimizing the entire architecture of a space transportation system [Hickman, 1998]. Note that all these applications of MDO were single criteria optimization problems.

In a few cases, MDO techniques have been investigated for their applicability to the design of satellites and, in one case, an entire environmental satellite constellation. The two organizations that have pioneered this work are the University of Colorado and The Aerospace Corporation.

The first, and to this date the most extensive, application of multidisciplinary design optimization to the design of a complete space system was performed by Dr. Mark Matossian at the University of Colorado [Matossian, 1995; Matossian, 1996]. Dr. Matossian employed a mixed integer linear programming model in combination with the branch-and-bound algorithm to find configurations for NASA's Earth Observation System (EOS) optimized with respect to scientific utility per unit cost. This was the first body of work that moved beyond the conventional satellite constellation design practice of optimizing strictly for coverage [Mason et al, 1998] or attempting to optimize for coverage and cost through trial and error, and deserves credit for pioneering the MDO approach on a complete space system. In this case, Dr. Matossian assumed a linear model for the system dynamics because it greatly simplifies the solution process by enabling the use of existing linear programming software that utilizes simplex and branch-and-bound algorithms to solve problems with thousands of decision variables and constraints. Similar software does not exist for nonlinear problems, which are more difficult to solve, even when they have fewer decision variables and constraints. Further, Dr. Matossian solved the EOS design problem as a single criteria (i.e. performance per unit cost) optimization problem rather than a more realistic multicriteria optimization problem. In spite of these simplifying assumptions, however, Dr. Matossian deserves credit for pioneering the application of MDO to satellite mission design.

After Dr. Matossian's work, the research in the field shifted away from the more difficult problem of optimizing the entire system and to the application of MDO to the design of an individual spacecraft. Now the scope of the system being optimized was a single satellite rather than an entire constellation. Dr. Todd Mosher was the first to survey several MDO techniques – including classical optimization, decomposition, Taguchi methods, and heuristics – for conceptual spacecraft design [Mosher, 1996]. Dr. Mosher eventually chose a heuristic approach using a genetic algorithm. He implemented this approach to create a software tool named SCOUT (Spacecraft Concept Optimization and Utility Tool) for the conceptual design of scientific spacecraft, and benchmarked the tool against NASA's Near Earth Asteroid Rendezvous (NEAR) spacecraft. The genetic algorithm successfully found

spacecraft bus designs – the decision variables were the solar cell type, solar array type, battery type, structure type, propulsion type, and the required launch vehicle – with a lower system (i.e. bus + launch vehicle) cost than the actual NEAR design [Mosher, 1998]. While the NEAR mission came within three percent of its initial projected cost and was a technical success, Dr. Mosher's work illustrated that even greater cost savings were possible. This work successfully demonstrated that it was possible to move away from the single design concept fixation historically used in space systems design. While the genetic algorithm worked well in this small example whose trade space contained only 192 design architectures, it was not computationally efficient as "families" of design solutions were continuously carried through and evaluated during the solution process. This computational inefficiency has led operations research scientists to prefer the use of other heuristics, such as simulated annealing, over genetic algorithms for certain classes of problems [Brooks et al, 1998].

Dr. Ellen Riddle explored the use of dynamic programming to provide insight into the effects individual technologies have on the performance and cost parameters of a satellite during the conceptual design phase [Riddle, 1998]. She developed a software tool named ESSAM (Early Small Satellite Analysis Method) employing a dynamic programming approach, and tested it on the design of a satellite power subsystem. ESSAM did find the optimal solution that minimized the power subsystem mass. However, this illustrative case study contained only two decision variables and four possible solutions. As previously stated, realistic space system design problems often contain many internally coupled decision variables, making the dynamic programming approach impractical for distributed satellite systems as dynamic programming can only handle problems of relatively small dimension.

MDO approaches have also been considered for application to spacecraft program management issues. Dr. David Bearden has used a genetic algorithm approach to create a software model named MERIT (Methodology for Evaluating, Ranking, and Inserting Technology) that balances risk and cost to aid in the development of a technology inser-

tion strategy for a satellite at the beginning of a mission [Bearden, 1999]. This tool takes into account the uncertainty in the actual evolution of a particular technology between the conceptual design of a spacecraft and the actual construction of the spacecraft, and the potential implications of this uncertainty on the total program cost.

As one can see, the bulk of space systems MDO work has been on individual spacecraft. The lone attempt to expand the scope of the design problem to an entire constellation maintained an assumption of linearity, an assumption which does not hold for distributed satellite systems due to the nonlinear couplings between the design variables and design metrics. Also, all of the space system design problems were assumed to be single criteria optimization problems in order to simplify their solution. Finally, none of this work included the time dimension of space systems – the tendency for individual components to fail and the impact these failures have on the performance of the system. The importance of capturing the effects of failures over time was recognized, but listed as "future work" by many of the above researchers. Thus, a new approach is needed to successfully integrate MDO techniques with the design of multicriteria distributed satellite systems taking into account the effect of time on system performance. MMDOSA will pioneer just such a new approach.

### **1.3.2 Space Systems Engineering Tools Review**

Several systems engineering tools exist for space systems design. Each of these tools was developed by different organizations with different uses in mind. Table 1.2 lists some of the major systems engineering tools in use today for space systems design. Notice that The Aerospace Corporation and the NASA Jet Propulsion Laboratory (JPL) have been the two pioneering organizations in the development of space systems engineering design tools. Both of these organizations do extensive work designing and evaluating Air Force and planetary exploration missions respectively. In fact, both organizations have dedicated design centers – the Conceptual Design Center (CDC) at The Aerospace Corporation and the Product Design Center (PDC) at JPL – to provide computer-aided analysis,

**TABLE 1.2** Survey of Systems Engineering Tools for Space Systems Conceptual Design [Bearden, 1998; Mosher, 2000]

<b>Tool</b>	<b>Organization</b>	<b>Use</b>
COBRA	The Aerospace Corporation	Automated assessment of program cost risk and schedule risk as a function of spacecraft complexity for interplanetary missions. [Bearden, 2000]
Concurrent Engineering Methodology (CEM)	The Aerospace Corporation	Mapping of "what if" cost and performance trade studies for Air Force missions. [Bearden & Law, 1996]
ESSAM	Univ. of Colorado	Small Satellite bus component selection. [Riddle, 1998]
GENSAT	Computational Technologies	Object-oriented software that interconnects existing commercial satellite subsystem tools (STK, CAD, IDEAS, etc.) and component databases for space systems design.
ICE	California Institute of Technology	Concept definition of novel space missions via integrated information systems. [Sercel et al, 1998]
MERIT	The Aerospace Corporation	Automated assessment of the cost and performance implications of inserting existing vs. new technologies into a spacecraft bus. [Bearden, 1999]
MIDAS	NASA Jet Propulsion Laboratory (JPL)	Analysis of proposed spacecraft designs via integrated tool executions on distributed machines. [Briggs, 1995; Fukunaga et al, 1997]
Modelsat	ROUTES	Cost and mass modeling for communications satellites. [Buckingham et al, 1995]
Project Trades Model (PTM)	JPL	Cost and performance prediction of novel interplanetary and space science missions. [Briggs, 1995]
QUICK	JPL	Spacecraft design programming language with extensive component databases and scaling relationships for conceptual spacecraft design. [Skinner, 1992]
SCOUT	The Aerospace Corporation	Single spacecraft mission bus component and launch vehicle selection. [Mosher, 1998]
SMALLSAT	NASA Langley Research Center	Earth observation spacecraft sensor and satellite bus configuration. [Greenberg, 1992]
SMAD	KB Sciences	Software automation of the calculations in Larson and Wertz's <u>Space Mission Analysis and Design</u> [KB Sciences, 1994]
SpaSat	Ball Aerospace	A preliminary spacecraft sizing, cost estimating, and orbital analysis tool for Ball Aerospace missions.

design, and trade studies for the missions these two organizations support. Also notice, however, that all of the tools listed in Table 1.2 are focused on the design of a single spacecraft rather than an entire distributed satellite system. The MMDOSA methodology

developed in this thesis will fill this niche for the conceptual design of distributed satellite systems.

## **1.4 Research Objectives and Approach**

### **1.4.1 Objectives**

The goal of this research is to demonstrate that distributed satellite system design problems can be formulated mathematically as optimization problems and solved to balance the system objectives and constraints in the conceptual design phase. This will be done within the framework of the GINA methodology, which is presented in Section 3.2. The solutions to these DSS conceptual design optimization problems will provide design teams with a better basis upon which to make future, more detailed system architecting and design decisions in the later development phases of the program. They will also provide an efficient mapping of the system trade space that will enable designers to focus their efforts on the region(s) of the trade space that will likely yield the best, most cost-effective architectures.

Specifically, this thesis develops a methodology and framework for mathematically modeling distributed satellite systems as optimization problems to enable an efficient search for the best (as defined by the metric(s) of interest to the customer) families of design solutions within the system trade space. The trade space for distributed satellite systems can be enormous – too large to enumerate, calculate, and compare all possible design architectures – due to the large number of design variables and the number of possible values for each design variable. The MMDOSA methodology will enable a greater search of the trade space and explore design architectures that might not otherwise be considered during the conceptual design phase. In doing so, the methodology will add structure to the conceptual design process by providing a means to move away from the previously discussed single point design fixation currently employed in space systems design. According to the International Council on Systems Engineering (INCOSE), 70%-90% of the development cost of a large system is predetermined by the time only 5%-10% of the

development time has been completed [INCOSE, 1998]. Thus, the ability to use MDO algorithms up front in the conceptual design phase will give systems engineers greater power to control lifecycle costs.

The specific objectives of the proposed research follow:

1. To *develop a framework and methodology* for mathematically formulating distributed satellite system conceptual design problems as optimization problems based on a solid theoretical foundation. The methodology will use MDO techniques to efficiently search the system trade space to find regions likely to contain the best solutions based on the metric(s) of choice. The solutions to the optimization problem will then guide the systems engineer as to where to focus efforts during the next phase of work.
2. To *integrate and couple* disparate capability, performance, reliability, and cost models to provide a full lifecycle analysis of distributed satellite systems.
3. To *advance the state-of-the-art* and develop specialized multiobjective optimization algorithms tailored to search the nonlinear, nonconvex trade spaces of distributed satellite systems.
4. To *document which MDO techniques and algorithms work best* for which types of distributed satellite systems. It is hypothesized that the best MDO approach will vary for different DSS instances and applications. For example, Taguchi methods with orthogonal arrays might work well for problems with small design vectors, but may become unmanageable for problems with large design vectors. Conversely, heuristic algorithms might work best for DSS problems with very large trade spaces, but may be inefficient for smaller problems. This information may be used to create "rules of thumb" dictating under which conditions each MDO approach is most applicable to the design of a distributed satellite system.
5. To *use the multiobjective, multidisciplinary design optimization systems architecting methodology* developed in this research *to find better design architectures* (based on the metric(s) of interest to the customer) than those that currently exist for each of the three mission case studies.

It should be pointed out that before MDO can be exercised, one must be able to link all of the different design tools (i.e. CAD, Matlab, Excel, Satellite Tool kit, etc.) that comprise the multidisciplinary models of the space system to be designed. This is an area of active research in both industry and academia, and several organizations are currently developing software that will provide a core common platform for interfacing different engineer-



ing software tools [Senin et al, 1999]. While ongoing research in cross-platform integration is relevant to the work presented in this thesis and may be linked to it in the future, it should be made clear that tool integration is not one of the objectives of this research. Rather, the goals are to develop, demonstrate, and evaluate an MDO methodology for the conceptual design of distributed satellite systems. All of the work presented here was executed in a Matlab [Mathworks, 2002] software environment that was cross-linked with Satellite Tool Kit [Analytical Graphics, 2002] as needed. While not as all encompassing as some of the other engineering software tools available in the market, Matlab does have several toolboxes that make it capable of handling a diverse range of technical engineering analyses, including finite element analysis, signal processing, and control design. These capabilities were sufficient for creating the multidisciplinary models required for the three case studies presented in this thesis.

### **1.4.2 Hypothesis**

This research is based on the hypothesis that the design of distributed satellite systems is a complex problem that must balance competing objectives and constraints. It is believed that improved system architectures will result from the application of MDO techniques to DSS conceptual design problems, provided that these problems can be properly modeled mathematically. This research is based on the belief that the necessary mathematical framework can be developed for DSS conceptual design problems, because such problems essentially are combinatorial problems in which the values for a large number of design variables must be selected in an attempt to minimize or maximize an objective(s) under a set of constraints.

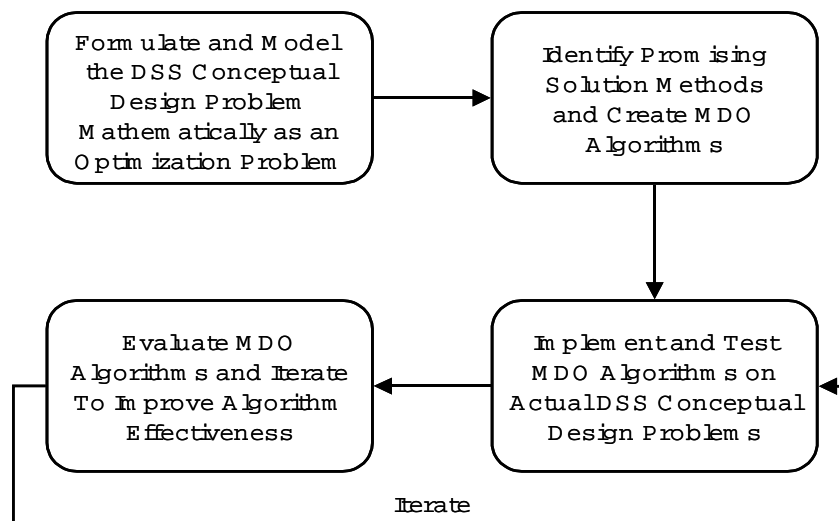
### **1.4.3 Approach**

This research represents a merger between multidisciplinary design optimization and space systems engineering to develop and evaluate a methodology for applying MDO techniques to the conceptual design of distributed satellite systems. This new methodology has a theoretical foundation in systems engineering and employs state-of-the-art

MDO algorithms that may be utilized by experts in industry and academia. The target application for the methodology is the conceptual design of distributed satellite systems, although the method should be applicable to a wider variety of space systems.

The approach in Figure 1.2 was followed to develop the MMDOSA methodology:

1. Create a framework and procedure for mathematically modeling and formulating the conceptual design of a DSS as an optimization problem.
2. Create MDO algorithms tailored to solve both single objective and multiobjective versions of the above DSS conceptual design optimization problem.
3. Apply these algorithms to three separate DSS conceptual design case studies and evaluate their effectiveness.
4. Statistically assess how well each MDO algorithm performs for each case study design problem, and identify patterns and lessons learned to be used by systems engineers in future DSS conceptual design exercises. Also use this information to iterate upon and improve the methodology.



**Figure 1.2** Research Approach

#### 1.4.4 Case Studies

The MMDOSA methodology is applied to three separate case studies (Table 1.3) representing each of the three sectors of space missions – civil, military, and commercial. The

Terrestrial Planet Finder, a civil mission, seeks to directly detect and characterize Earth-like planets orbiting neighboring stars within 15 parsecs of Earth. This case study illustrates the power of the MMDOSA methodology to find the best families of solutions (i.e. most cost-effective architectures) for the single objective DSS design problem and introduces the two-dimensional multiobjective design problem. TechSat 21, a military mission, will demonstrate the ability of a cluster of satellites to work collaboratively to detect and track ground-based targets. This case study illustrates the ability of the MMDOSA methodology to handle higher-order multiobjective DSS design problems, and to find the Pareto optimal set of system architectures for such problems. Finally, broadband communications, a commercial mission, seeks to provide high data rate services to system subscribers. This case study illustrates the flexibility of the MMDOSA methodology by analyzing different business case scenarios under uncertain market conditions.

**TABLE 1.3** MMDOSA Case Studies

<b>Name</b>	<b>Sector</b>	<b>Sponsor</b>	<b>Mission</b>
Terrestrial Planet Finder	Civil	NASA	Planet Detection and Characterization
TechSat 21	Military	AFRL	Ground Moving Target Indication
Broadband	Commercial	Corporation	High Data Rate Communication Services

These three case studies serve as experiments in which the MMDOSA methodology is applied to a conceptual design problem that is identical to one recently studied by various aerospace organizations. The architectures identified by the MMDOSA methodology are compared with the architectures proposed by the various organizations to see if the MMDOSA methodology does indeed lead to better conceptual designs. The primary metrics for better are defined by each customer's goals, and may be generally characterized as providing improved projected mission performance with a lower projected lifecycle cost. In these conceptual design experiments, the flexibility of the MMDOSA methodology may also be tested. This will be done by changing a key requirement and 1) seeing how well or how poorly the current design architecture handles that requirement and 2) quickly

searching the design space to find families of solutions that best meet the new requirement.

## **1.5 Thesis Overview**

The remainder of this document develops and demonstrates the multiobjective, multidisciplinary design optimization systems architecting methodology (MMDOSA) for the conceptual design of distributed satellite systems. Chapter 2 discusses how MMDOSA fits into the space systems design process, defines the mathematical optimization foundation on which this research is based, and formulates the DSS conceptual design problem as an optimization problem. Chapter 3, the crux of this thesis, details the MMDOSA methodology step by step. Chapters 4-6 illustrate and evaluate the application of MMDOSA to the TPF, TechSat 21, and broadband case studies respectively. Finally, Chapter 7 states the conclusions from this research, the contributions this work has made to the field of space systems engineering, and recommendations for future work.

# Chapter 2

## PROBLEM FORMULATION

This chapter formulates the distributed satellite system conceptual design problem as a mathematical optimization problem within the context of formal systems engineering principles. First, explicit definitions are given for different systems engineering terms. Next, the operations research principles on which the optimization formulation is based are presented. Finally, the DSS conceptual design problem is formally modeled as an optimization problem.

### 2.1 Systems Engineering Principles

#### 2.1.1 The Space Systems Design and Development Process

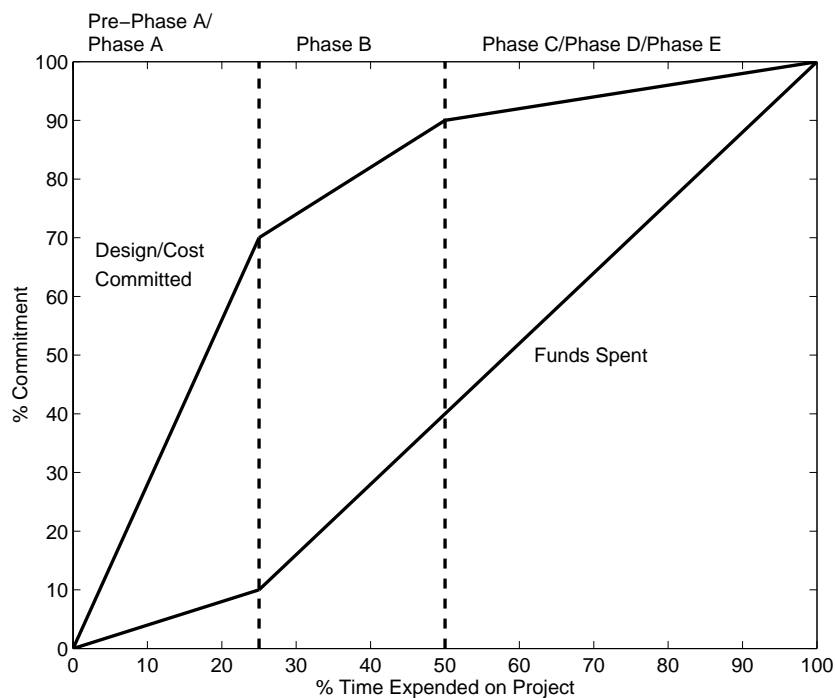
To understand how the MMDOSA methodology can contribute to the conceptual design of space systems, one must first understand the process by which such systems are designed and developed. Most space programs follow a six phase design and development cycle [Shishko & Chamberlain, 1995].

- Pre-Phase A: Conceptual Study
- Phase A: Preliminary Analysis
- Phase B: Definition
- Phase C: Design
- Phase D: Development
- Phase E: Operations

Pre-Phase A work entails the translation of mission ideas into mission concepts. This includes the development of preliminary requirements, determination of evaluation metrics, creation of alternative mission architectures, preliminary analysis and trades of these architectures, and initial cost estimation. Phase A work involves more detailed trade analysis of the alternative mission architectures, development of initial risk management strategies, and refining of the initial cost estimates. Phase A results in the identification of the best design architecture(s) from the analysis of all alternative architectures. Phase B takes the preliminary design architecture(s) and defines these designs in greater technical detail to establish an initial baseline capable of meeting the mission needs. This phase includes the development of a systems engineering management plan, a risk management plan, a work breakdown structure, and a concept of operations in preparation of the actual construction and flight of the mission. Phase C completes the detailed technical design of the mission, including the final requirements document, all lower-level design specifications, interface control, and the manufacturing plan. Phase C concludes the design of the space system. In Phase D, the system is actually built. This phase includes fabrication, integration, testing, launch, and orbital check out. Finally, Phase E entails execution of the mission, including operations, maintenance, and disposal.

The MMDOSA methodology presented in this thesis has been developed for use during Pre-Phase A and Phase A work. During these early stages of conceptual design, the designer has the opportunity to generate, analyze, and compare a vast array of architectures for the given mission. Unfortunately, as discussed in Section 1.3.2, not many systems engineering tools currently exist for the lifecycle analysis of distributed satellite systems. This leads to the point design fixation approach. The risk exists in such an approach that more cost-effective architectures that can meet all of the mission requirements will never be considered. Figure 2.1 plots the time expended vs. funds/lifecycle cost committed for a typical space mission. Even though Pre-Phase A/Phase A conceptual work represents only 25% of the program man-hours and 10% of the budget, the decisions made during these early stages lock in a majority of the lifecycle cost. In fact, INCOSE estimates that 70%-90% of the development cost of a large system is predetermined by the

time only 5%-10% of the development time has been completed [INCOSE, 1998]. Thus, better systems engineering tools are needed to enable a more intelligent, thorough search of the trade space during the conceptual design phases of a program, where the majority of the design and lifecycle costs are locked in. MMDOSA will provide just such a tool to help systems engineers make better up-front design choices to improve the system performance and reduce the lifecycle cost of future distributed satellite systems.



**Figure 2.1** Time Expended vs. Funds/Cost Committed for a Typical Space Project [Casani, 1994]

### 2.1.2 Systems Engineering Definitions

This section explicitly defines several terms that are used repeatedly throughout this thesis, and places these terms within the context of the MMDOSA methodology. INCOSE defines a *system* as an integrated set of elements that work together to achieve a defined objective [INCOSE, 1998]. The elements within a system can include hardware, software, people, information, objects, processes, and facilities. The type of system this research focuses on is a distributed satellite system – a system of multiple satellites that

work together in a coordinated fashion to perform a mission [Shaw et al, 2000]. *Systems engineering* is defined as an interdisciplinary practice that enables the realization of successful systems [INCOSE, 1998]. The focus is on the system as a whole rather than on the individual components of the system, as the individual components and subsystems do not necessarily need to be individually optimized for the system to perform optimally. Rather, optimal performance results from the synergistic integration of the components. MMDOSA is intended to be a tool to improve the systems engineering of distributed satellite systems with the goal of improving the cost-effectiveness of space systems.

*Systems analysis* is the process of helping decision makers choose preferred courses of action by 1) systematically examining the relevant objectives and strategies for achieving these objectives and 2) comparing quantitatively the effectiveness, costs, and risks of the alternatives [Shishko & Chamberlain, 1995]. Both GINA and MMDOSA are systems analysis tools for distributed satellite systems. *Functional analysis* is the process of identifying, describing, and relating the functions a system must perform to fulfill its goals and objectives [Shishko & Chamberlain, 1995]. The GINA methodology, described in Section 3.2, is also a functional analysis tool for distributed satellite systems.

A *system architecture* is defined as the structure, arrangement, or configuration of a system of elements and the relationships required to satisfy both the constraints and a set of functional, performance, reliability, maintainability, and extensibility requirements [Boppe, 1997]. As will be explained in Section 2.3.1, a given set of values for the elements within a design vector of a GINA model defines a unique architecture for a DSS. *Systems architecting* is defined as the process of creating a system architecture that satisfies the mission requirements while concurrently optimizing the system within a set of budget, schedule, and risk constraints [INCOSE, 1998]. MMDOSA is a systems architecting methodology. Finally, *trade studies* provide a means for comparing the relative merit of alternative architectures. GINA provides a means of executing trade studies for a DSS, and MMDOSA uses the results from such trade studies to automatically search for optimal system architectures.





Each  $x_i$  is a *decision variable* in the optimization problem. The decision variables are the parameters whose values may be varied over a given range to minimize or maximize the desired function, and may be real continuous, integer, discrete, or binary. The  $c_i$  terms are the *cost coefficients* associated with each of the decision variables. The  $c_i$  and  $x_i$  terms combine to create the *objective function* to be minimized or maximized. The next three lines in Eqn. 2.1 represent a set of *constraint equations* which place limits  $b_j$  on certain combinations of the decision variables. Constraint equations may exist as both inequality and equality constraints. The final two lines represent the upper  $u_i$  and lower  $l_i$  *bounds* placed on each of the decision variables. When the lower bounds on the decision variables are set to zero, these constraints are termed *non-negativity constraints*.

A *feasible solution* is a combination of values for the decision variables  $x_i$  that satisfies all of the constraint equations as well as the upper and lower bounds on the decision variables. An *infeasible solution* is a combination of values for the decision variables  $x_i$  that violates one or more constraints. An *optimal solution* is the combination of values for the decision variables  $x_i$  that comprises a feasible solution and minimizes or maximizes the objective function. In Section 2.3, these components of a classical optimization problem will be mapped onto the formulation of the DSS conceptual design problem as a mathematical optimization problem.

### **2.2.2 Trade Space Convexity – Local Optima vs. Global Optima**

Convexity and concavity play important roles in optimization. A *convex* function is a function that cannot have a local minimum that fails to be a global minimum [Bertsimas & Tsitsiklis, 1997]. Similarly, a *concave* function is a function that cannot have a local maximum that fails to be a global maximum. A solution is *locally optimal* if there is no neighboring solution that has a better objective function value. A solution is *globally optimal* if there is no other solution in the entire feasible trade space that has a better objective function value. These definitions apply to single objective optimization problems.

The function in Eqn. 2.2 is a convex function. The local minimum at  $x=0$  is also the global minimum (Figure 2.2a).

$$f(x) = x^2 \quad (2.2)$$

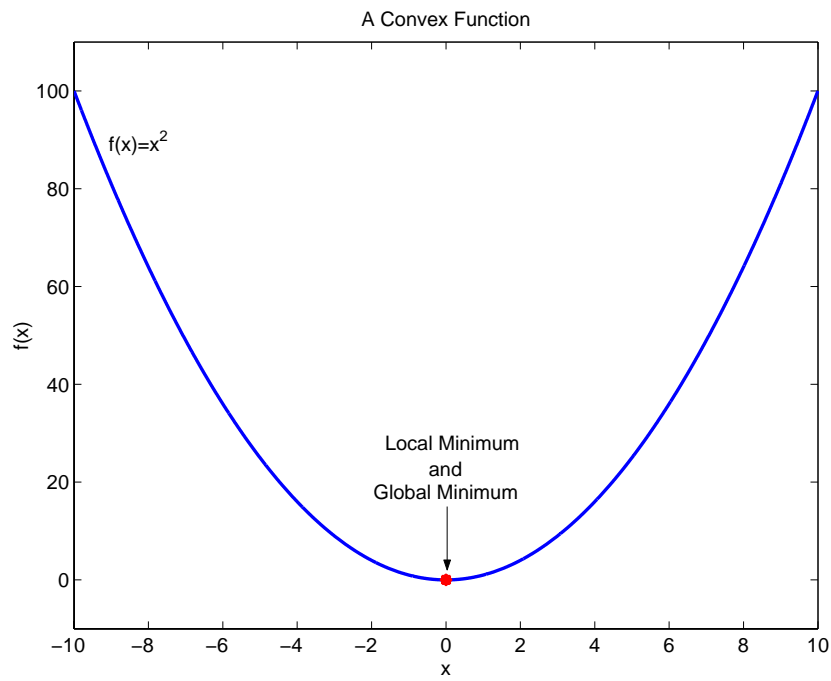
Similarly, the function in Eqn. 2.3 is a concave function in which the local maximum at  $x=0$  is also the global maximum (Figure 2.2b).

$$f(x) = 10 - x^2 \quad (2.3)$$

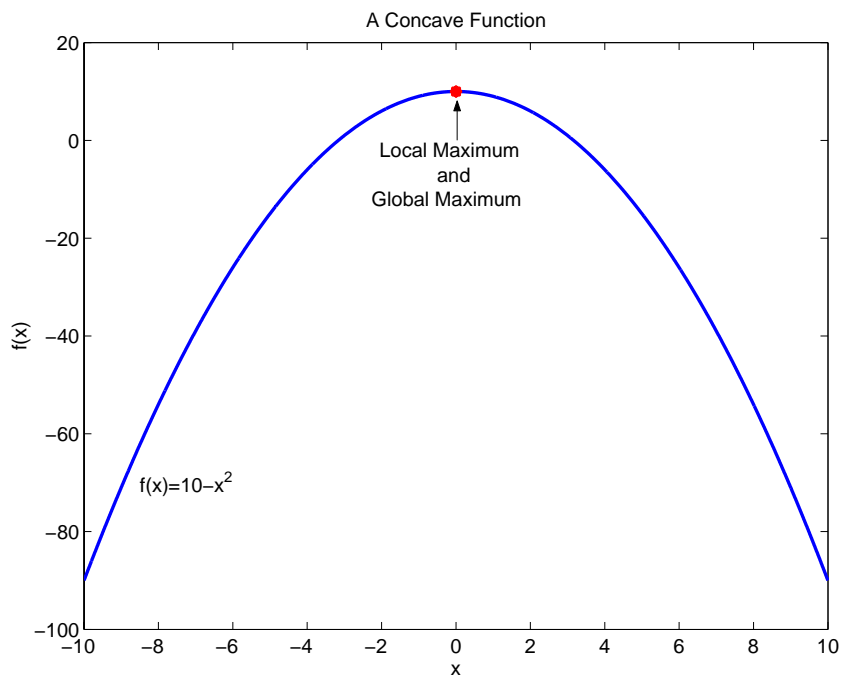
Conversely, local minima that are not global minima may exist in a *nonconvex* function. The function in Eqn. 2.4 is a nonconvex function (Figure 2.3). This function contains many local optima that are not globally optimal. An algorithm searching for the global minimum might converge on one of the local optima and incorrectly conclude that it is the global optimum.

$$f(x) = \cos(14.5x - 0.3) + (x + 0.2)x \quad (2.4)$$

These same principles apply to a DSS trade space with any number of dimensions. If the trade space is convex or concave, it should be fairly straightforward to find the optimal design architecture for a single objective optimization problem using gradient-based techniques. However, if the trade space is neither convex nor concave, then the danger exists that the search algorithm may become trapped in a local optimum. This in turn might lead the systems engineer to incorrectly conclude that a particular system architecture merits further study when in fact there exist alternative architectures in the global trade space that perform far better based on the metric(s) of interest to the customer. Thus, care must be taken to avoid local optima when designing algorithms to search the trade space of a distributed satellite system. The algorithms incorporated into the MMDOSA methodology have been selected and designed with this in mind, and are detailed in Section 3.5.

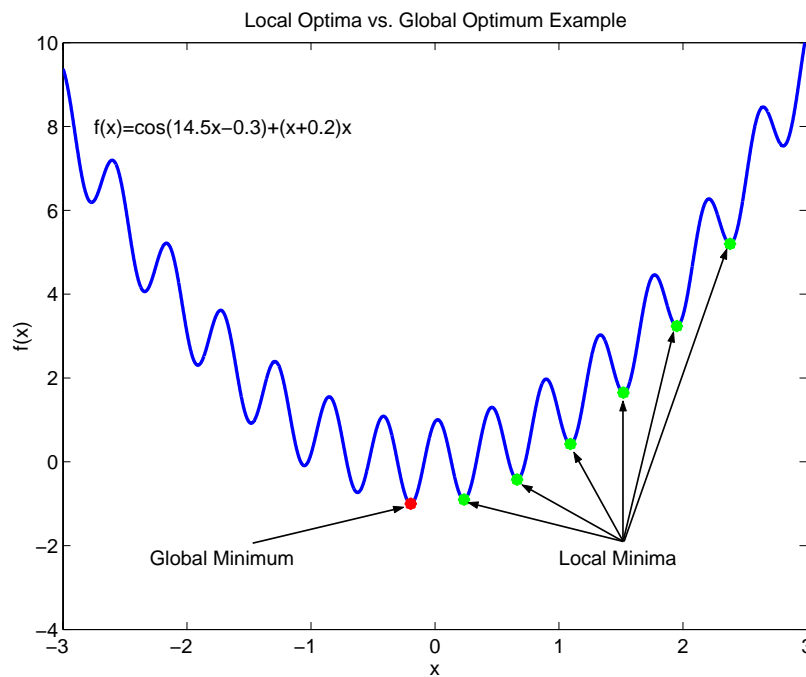


(a)



(b)

**Figure 2.2** Example Convex (a) and Concave (b) Functions



**Figure 2.3** Example Nonconvex Function

### 2.2.3 Optimization Problem Classification

Optimization problems may be classified according to the structure of the objective function, the nature of the decision variables, the structure of the model, the existence and type of constraints, the nature of the parameters, and the convexity of the trade space. Table 2.1 lists these classifications. This table can help one understand the structure of the DSS conceptual design optimization problem that is formally defined in Section 2.3. Some DSS conceptual design problems seek to optimize the system architecture with respect to a single metric (i.e. minimize lifecycle cost), while other DSS conceptual design problems will simultaneously attempt to optimize multiple criteria (i.e. minimize lifecycle cost and maximize system performance). The decision variables in a DSS design problem can be real continuous (i.e. any orbital altitude between 100 km and 1000 km), real discrete (i.e. any aperture diameter between 0.5 m - 4.0 m in increments of 0.5 m), integer (i.e. any number of satellites per plane between one and eight), binary (i.e. whether or not to adopt intrasatellite cross-links for communication), or any combination of these. All of

the decision variables in the DSS design problems investigated in this research are static variables (i.e. they do not change dynamically over time). The multidisciplinary models used to simulate the system dynamics contain both linear (i.e. the mean number of satellites in view as a function of the number of satellites per orbital plane) and nonlinear (i.e. learning curve savings resulting from mass production of identical satellite buses) relationships, but are dominated by the nonlinear relationships. All of the DSS conceptual design problems considered were constrained optimization problems with both deterministic and stochastic parameters and nonconvex/nonconcave trade spaces.

**TABLE 2.1** Optimization Problem Taxonomy

<b>Objective Function Structure</b>	<b>Decision Variable Type</b>	<b>Decision Variable Nature</b>	<b>Model Structure</b>	<b>Constraint Existence</b>	<b>Parameter Nature</b>	<b>Trade Space Convexity</b>
Single Objective	Real Continuous/Discrete	Static	Linear	Unconstrained	Deterministic	Convex/Concave
Multiple Objectives	Integer/Binary	Dynamic	Nonlinear	Constrained	Stochastic	Nonconvex/Nonconcave

The DSS conceptual design problem may be further categorized as a combinatorial, or vector, optimization problem. In a combinatorial problem, there exist many combinations of the design variables that lead to feasible solutions. Combinatorial optimization problems contain a finite, but often very large number of feasible solutions. While easy to formulate, combinatorial problems are often very hard to solve to optimality. Such problems are referred to as NP-hard or NP-complete problems [Bertsimas & Tsitsiklis, 1997].

## 2.2.4 Solution Techniques

Several solution techniques exist for each type of optimization problem. The classical technique for linear programming problems (i.e. a problem with a linear objective function and constraint equations) is the simplex method. If the formulation of a linear program contains a special structure, network flow, ellipsoid, or interior point methods may be also be used [Bertsimas & Tsitsiklis, 1997]. For problems whose decision variables are

restricted to integer values, known as integer programs, branch and bound, cutting plane, and dynamic programming methods may be employed to search for the optimal solution [Bertsimas & Tsitsiklis, 1997]. Stochastic heuristic techniques – including genetic algorithms, simulated annealing, and tabu search – work best on combinatorial problems with nonconvex trade spaces [Aarts & Lenstra, 1997]. Unlike other methods, heuristic techniques are not guaranteed to find global optima. Unfortunately, the DSS conceptual design problem cannot be formulated as either a linear program (i.e. contains nonlinear models and objective functions) or an integer program (i.e. not all of the decision variables are restricted to integer values), and thus the powerful algorithms that exist for solving such problems cannot be used. As explained in Section 2.2.3, empirical evidence shows that DSS conceptual design problems are combinatorial nonlinear problems with nonconvex trade spaces. In Section 3.5, it will be shown that, as a result of the problem structure, heuristic techniques work best on DSS conceptual design optimization problems.

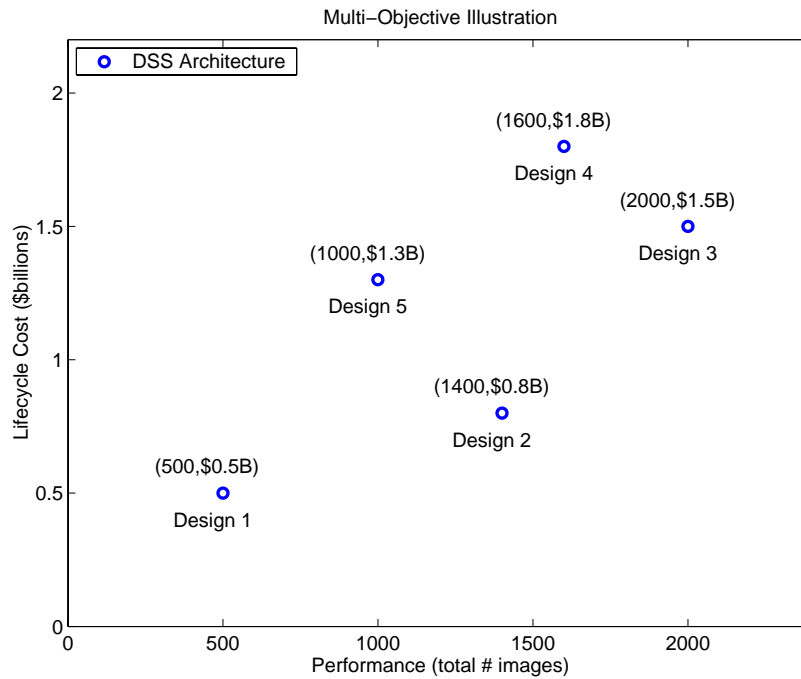
### 2.2.5 Single Objective Optimization vs. Multiobjective Optimization

The differences between single objective and multiobjective optimization problems are directly relevant to this research and will be introduced through an example DSS conceptual design problem – the design of a separated spacecraft telescope. Figure 2.4 plots the performance versus lifecycle cost for five different architectures of the space telescope system. Table 2.2 lists the key parameters for each architecture.

#### Single Objective Optimization

A single objective optimization problem is a problem in which one seeks the best (i.e. highest or lowest) value of a well-defined objective [Goldberg, 1989]. Three possible single objective functions for the example DSS optimization problem are listed in Eqn. 2.5.

$$\begin{aligned}
 &\text{Minimize}(\textit{Lifecycle Cost}) \\
 &\text{Maximize}(\textit{System Performance}) \\
 &\text{Minimize}\left(\frac{\textit{Lifecycle Cost}}{\textit{System Performance}}\right)
 \end{aligned}
 \tag{2.5}$$



**Figure 2.4** Five Theoretical Separated Spacecraft Telescope Design Architectures

**TABLE 2.2** Key Parameters of Five Separated Spacecraft Telescope Architectures

Design	Lifecycle Cost (\$B)	Performance (# Images)	Cost Per Image (\$M)
1	0.5	500	1.00
2	0.8	1400	0.57
3	1.5	2000	0.75
4	1.8	1600	1.13
5	1.3	1000	1.30

If the problem is convex for a minimization objective function or concave for a maximization objective function, there will exist only one true optimal solution to the problem. If the problem is nonconvex/nonconcave, there may exist more than one globally optimal solution, but each globally optimal solution will have the same objective function value.

Table 2.3 lists the solution for each objective function in Eqn. 2.5 to the sample problem in Figure 2.4. Notice how the optimal system architecture changes when the objective func-



tion in the problem formulation changes. Thus, in applying the MMDOSA methodology to single objective DSS conceptual design problems, the algorithm will be searching for a single optimal solution to the mathematical problem at hand. Even if the algorithm does not find the true optimal solution, it should find some of the families of best solutions. Section 3.5.1 introduces procedures and algorithms for solving the single objective optimization version of the DSS conceptual design problem.

**TABLE 2.3** Solutions to the Separated Spacecraft Telescope Design Problem

Objective Function	Solution	Optimal Objective Function Value
Minimize( <i>Lifecycle Cost</i> )	Design 1	\$0.5B
Maximize( <i>System Performance</i> )	Design 3	2000 Images
Minimize( <i>Cost Per Image</i> )	Design 2	\$0.57M/Image

### Multiobjective Optimization

While single objective optimization provides a powerful tool to explore the trade space of a DSS, there are times when several important decision criteria are present simultaneously and it is not possible to combine these criteria into a single number [Goldberg, 1989]. These problems are classified as multiobjective, or multicriteria, optimization problems – problems in which one attempts to optimize several separate criteria simultaneously. Such problems are important to the DSS conceptual design problem because true systems methods handle *trades* between multiple objectives, not just a single objective function. This is because in real systems engineering problems, one has to *balance* multiple requirements while trying to simultaneously achieve *multiple* goals. Therefore any MDO methodology for DSS conceptual design needs to be able to handle multiple objective functions.

Several differences exist between single objective and multiobjective optimization problems. Convex/concave single objective problems have only one true optimal solution – there exists a single optimal value to minimize or maximize the objective function. Multiobjective problems, however, can have several optimal solutions. At this point, it

becomes necessary to define some terminology, which will be done within the context of the example in Figure 2.4.

Keeping in step with the separated spacecraft telescope design problem, now let performance and lifecycle cost be separate objective functions. An ideal system design will maximize system the total performance while *simultaneously* minimizing lifecycle cost.

$$\text{Maximize}(\textit{System Performance}) \text{ AND } \text{Minimize}(\textit{Lifecycle Cost}) \quad (2.6)$$

Suppose one is considering the five system architectures in Figure 2.4, each with a different performance and lifecycle cost (Table 2.2). At first glance, it is not clear which design architecture is optimal according to the objective function in Eqn. 2.6. The notion of optimality in multiobjective problems is often not as obvious as in single objective problems. Recalling that the goal is to simultaneously achieve the objectives of maximizing system performance and minimizing lifecycle cost, the best designs are located towards the lower right corner of the plot. Thus, Designs 1, 2, and 3 appear to be the best choices. However, none of these three designs is the best along both dimensions. In other words, there exist *trade-offs* between each of these three system architectures. For example, moving from Design 1 to Designs 2 or 3 improves the performance metric, but at the *expense* of the lifecycle cost metric – both metrics can't be improved simultaneously. In optimization terminology, these three architectures are defined to be *nondominated*, or *non-inferior*, because there exist no better architectures than these for all decision criteria [Goldberg, 1989].

On the other hand, Designs 4 and 5 are poor choices because both of these architectures are dominated by other architectures within the trade space. A dominated solution is a solution which is worse than at least one other solution in the trade space for all decision criteria. Design 4 is dominated by Design 3 because 1600 images < 2000 images and \$1.8B > \$1.5B. Likewise, Design 5 is dominated by Design 2 as 1000 images < 1400 images and \$1.3B > \$0.8B. Architectures 4 and 5 are clearly suboptimal, and are classified as *dominated*, or *inferior*, architectures.

A definition for multiobjective optimality that respects the integrity of each of the separate decision criteria may now be developed. The formal definition of optimality for multiobjective problems stems from the work of Vilfredo Pareto, an economics Professor at the University of Lausanne in Switzerland who in the early 1900's developed a theory for the optimal allocation of resources in a society with competing objectives [Pareto, 1906]. The notion of Pareto optimality – the optimal solution to a multiobjective problem – stems from this work and in the words of Pareto himself may be defined as the following:

The optimum allocation of the resources of a society is not attained so long as it is possible to make at least one individual better off in his own estimation while keeping others as well off as before in their own estimation. – *Vilfredo Pareto* [Pareto, 1906]

A more modern definition of multiobjective optimality comes from the Dictionary of Cybernetics and Systems:

The solution to a multiobjective problem is, as a rule, not a particular value, but a set of values of decision variables such that, for each element in this set, none of the objective functions can be further increased without a decrease of some of the remaining objective functions. All such values for the decision variables that satisfy this condition are referred to as Pareto optimal. – *Dictionary of Cybernetics & Systems* [Principia Cybernetica, 2002]

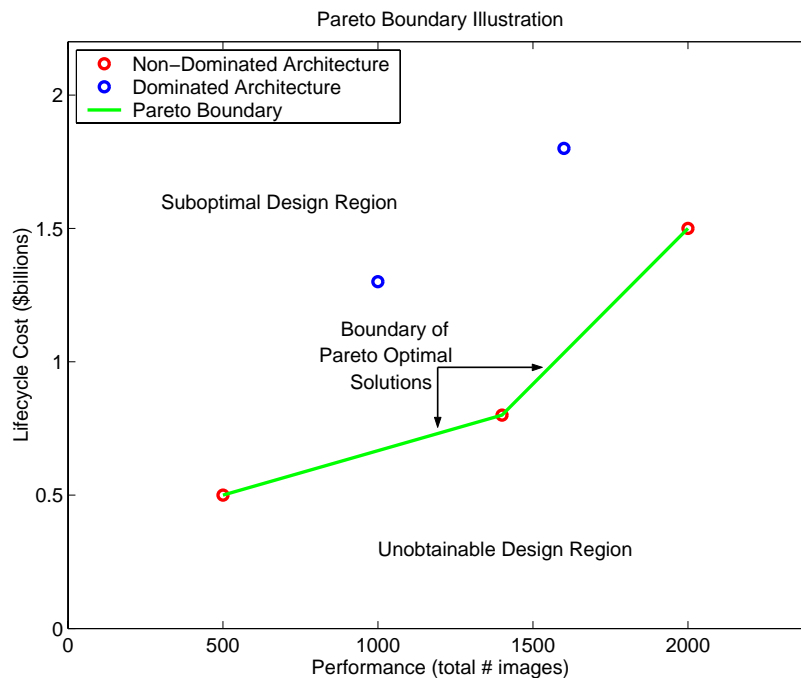
In this example problem, as in all multi-criteria problems, instead of obtaining a single answer, a set of solutions {Design 1, Design 2, and Design 3} have been obtained that are not dominated by any other solutions – the Pareto optimal (P-optimal) set. This is a key point – multiobjective problems can have more than one solution while single objective problems have only one true solution. Pareto optimality does not tell the systems engineer which system architecture is best, but rather provides the systems engineer with a subset of the most efficient design solutions within the vast trade space.

Let  $X$  represent the entire trade space of a multiobjective problem with  $j$  objective minimization functions ( $O_j$ );  $X$  contains  $i$  total points, and each point can be represented as  $x_i$ . Point  $y$  is mathematically said to be Pareto optimal when there exists no other point in  $X$  which meets the following condition:

$$O_j(x_i) < O_j(y) \quad \text{for all } i \text{ and } j \quad (2.7)$$

The group of points within  $X$  that meet this condition comprise the Pareto optimal set.

This simple example considered a DSS conceptual design problem with only two objective functions – two decision criteria and two dimensions. When plotting the results from such a two-dimensional problem, the Pareto optimal set creates a boundary of the most efficient architectures within the trade space (Figure 2.5). However, the same approach, definitions, and formula hold for a multiobjective problem with any number of decision criteria – any number of dimensions. For example, in addition to maximizing system performance and minimizing lifecycle cost in the separated spacecraft telescope case study, one may also wish to maximize system reliability, minimize the size of the operations crew, maximize the flexibility of the system, or do any combination of these items. Each of these objective functions add another dimension to the problem, but the concepts and methods of Pareto optimality remain the same.



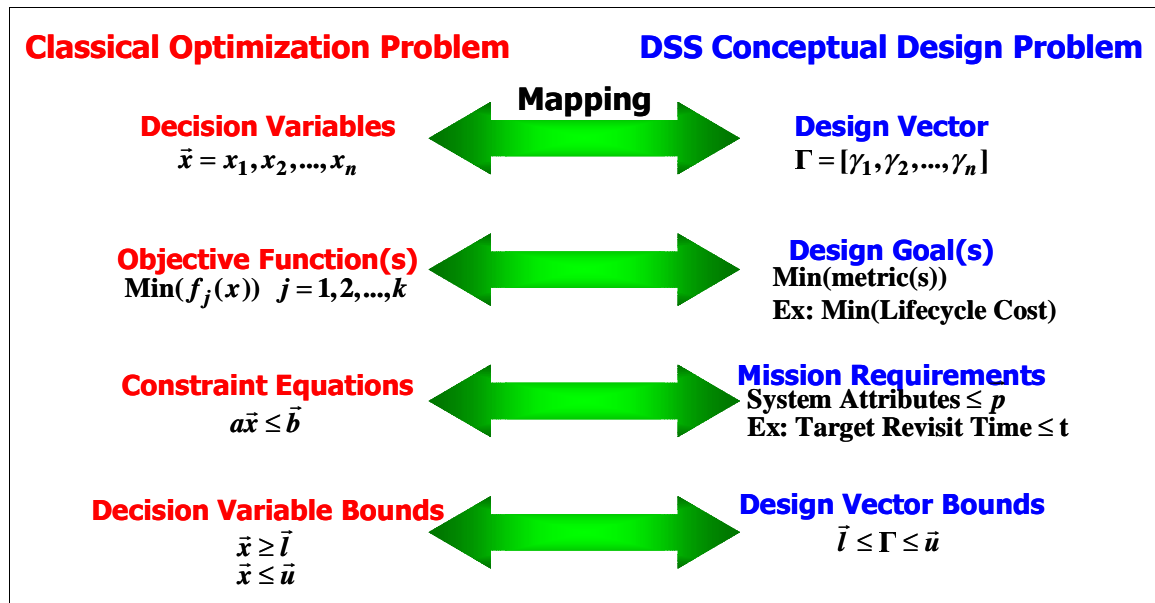
**Figure 2.5** The Pareto Boundary in a Two-Dimensional Trade Space

To summarize, multiobjective problems attempt to simultaneously optimize multiple criteria. In doing so, such problems do not yield a single solution, but rather a set of P-optimal solutions. Because this problem formulation more accurately represents real world DSS conceptual design problems, determining the Pareto optimal set within the vast DSS design trade space will become the focus of the MMDOSA methodology in Section 3.5.2.

## 2.3 DSS Conceptual Design Problem Formulation

### 2.3.1 The Standard Formulation

Recall that a classical optimization problem (Eqn. 2.1) contains decision variables, an objective function, constraint equations, and decision variable bounds. Figure 2.6 maps these elements of a classical optimization problem onto the distributed satellite system conceptual design problem. By examining this figure, one can begin to understand the transformation of a DSS systems engineering conceptual design problem into a mathematical optimization problem.



**Figure 2.6** Transformation Between the Classical Optimization Problem and the DSS Conceptual Design Problem

The decision variables in the classical optimization problem are the parameters whose values may be varied over a given range to minimize or maximize the desired function. Similarly, the design vector  $\Gamma$  in a DSS conceptual design problem contains the key independent design parameters the systems engineer has control over in developing a system architecture to accomplish the mission. Sample parameters  $\gamma_i$  that may be placed in the design vector include the number of satellites in the system, the orbital altitude of the satellites, and so forth. A unique set of values in the design vector defines a unique system architecture. Thus, the design vector represents the set of design variables with which the DSS architecture may be optimized. The remaining design parameters are placed into a constants vector. The constants vector  $K$  contains parameters which influence the design of the architecture, but which are held constant when comparing different system architectures. For example, the areal density of the satellite solar arrays is a parameter which will partly determine the mass of each satellite, which in turn will determine which launch vehicle(s) will be required to deploy the system, which in turn affects the lifecycle cost of the system, etc. However, if the solar array areal density is held constant for architectures with different numbers of satellites at different orbital altitudes, this parameter belongs in the constants vector. Each design input parameter is placed in either the design vector or constants vector at the discretion of the systems engineer, and these parameters may be transferred between vectors during the conceptual design process.

In a classical optimization problem, the decision variables combine with the cost coefficients to create the objective function to be optimized (i.e. minimized or maximized). In a DSS conceptual design problem, the design goal(s) important to the customer are optimized. Such goals may be to maximize system performance, minimize lifecycle cost, maximize system reliability, etc. Herein lies an important difference between classical optimization problems and the DSS conceptual design problem. While the objective function in a classical optimization problem may be written and evaluated as a direct closed form relationship between the decision variables and cost coefficients, no such closed form analytical relationship exists between the variables in the design vector and the design goals in the DSS conceptual design problem. Rather, the relationship between the

design vector and the mission goals may be thought of as a "black box." The black box is actually the GINA model (developed in Section 3.2) that relates the input design and constants vectors to the output design metrics and attributes via a sequence of mission models (orbital dynamics, payload physics, system launch, operations, etc.). In other words, the black box is a simulation of a single architecture of the distributed satellite system. As discussed in greater detail in Section 3.5, this black box characteristic of DSS conceptual design problems affects the selection of design optimization solution methods in the MMDOSA methodology.

The constraint equations in a classical optimization problem place limits on certain combinations of the decision variables – combinations of the decision variables that violate one or more constraint equations represent infeasible solutions. Likewise, design requirements place limits on the allowable values for different mission attributes – system architectures which violate one or more of the design requirements represent unacceptable architectures. Sample DSS design requirements might be a maximum allowable target revisit time, a minimum required SNR for a given payload measurement, etc. Every mission requirement can be represented as a constraint equation in the problem formulation. Every architecture that is explored by the optimization algorithm will be checked against the entire set of constraint equations – against the entire set of mission requirements. If any of the constraint equations are violated (i.e. if any of the mission requirements are not met), then that system architecture is classified as an infeasible solution and the algorithm proceeds to a new design. Again, while there exists a closed form analytical relationship between the decision variables and the constraint equations in the classical optimization problem, there exists no such analytical relationship in the DSS conceptual design problem. Rather, a black box – the GINA model of the distributed satellite system – relates the decision variables in the design vector to the design requirement constraints.

The final set of constraint equations in the classical optimization problem place upper and lower bounds on each of the decision variables. Similar bounds may be placed on each of

the design variables in the design vector. Furthermore, these DSS design variables may be discretized to reduce the size of the feasible trade space.

### 2.3.2 Sample DSS Conceptual Design Problem Formulation

In this section, the mathematical formulation of an actual DSS conceptual design problem as an optimization problem using the mapping defined in the previous section (Figure 2.6) is presented. This is done for a proposed space-based radar constellation named TechSat 21. Chapter 5 presents the detailed application of the MMDOSA methodology to the TechSat 21 mission, including modeling, optimization, and sensitivity analysis. The TechSat 21 conceptual design problem contains six key decision variables which comprise the design vector  $\Gamma$  in Eqn. 2.8. These variables are defined in Table 2.4. Notice that the design trade space contains over half a million different system architectures.

$$\Gamma = [\gamma_1 \ \gamma_2 \ \gamma_3 \ \gamma_4 \ \gamma_5 \ \gamma_6] \quad (2.8)$$

**TABLE 2.4** The TechSat 21 Design Vector

<b>Design Vector Variable</b>	$\Gamma$	<b>Range</b>	<b>Discretization</b>
# Satellites Per Cluster	$\gamma_1$	4 - 16	1
Aperture Diameter	$\gamma_2$	0.5 m - 4.0 m	0.5 m
Radar Transmission Power	$\gamma_3$	100 W - 1000 W	100 W
Constellation Altitude	$\gamma_4$	500 km - 1500 km	100 km
# Clusters Per Orbital Plane	$\gamma_5$	3 - 10	1
# of Orbital Planes	$\gamma_6$	3 - 10	1
<b>Total # Permutations</b>		<b>732,160</b>	

Assume the customer of this system, the U.S. Air Force, desires to field a constellation that minimizes a cost per unit probability of detection metric, provides global coverage, and has a maximum theater revisit time no greater than 15 minutes. The mathematical formulation of this single objective TechSat 21 conceptual design problem as an optimization problem is given in Eqn. 2.9.



$$\begin{aligned}
 \text{Objective:} & \quad \text{Min} \left( \frac{\sum_{y=1}^{10} \Phi_y(\Gamma)}{\frac{\sum_{y=1}^{10} \Psi_{Pd_y}(\Gamma)}{10}} \cdot (100) \right) \\
 \text{Constraints:} & \quad \text{Subject to} \\
 & \quad \sigma_T \leq 10 \text{ dbsm} \\
 & \quad MDV \leq 1.7 \text{ m/s} \\
 & \quad \Psi_{Pd_y} \geq 0.95 \text{ for all } y \tag{2.9} \\
 & \quad T_{tr} \leq 15 \text{ min} \\
 & \quad T_{tu} \leq 1 \text{ min} \\
 & \quad A_v \geq 0.95 \\
 & \quad 4 \leq \gamma_1 \leq 16 \\
 & \quad 0.5 \text{ m} \leq \gamma_2 \leq 4.0 \text{ m} \\
 & \quad 100 \text{ W} \leq \gamma_3 \leq 1000 \text{ W} \\
 & \quad 500 \text{ km} \leq \gamma_4 \leq 1500 \text{ km} \\
 & \quad 3 \leq \gamma_5 \leq 10 \\
 & \quad 3 \leq \gamma_6 \leq 10
 \end{aligned}$$

where  $y$  is the year of the mission,  $\Phi$  is the system cost,  $\Psi_{Pd}$  is the target probability of detection,  $\sigma_T$  is the minimum target radar cross section,  $MDV$  is the target minimum detectable velocity,  $T_{tr}$  is the theater revisit time,  $T_{tu}$  is the target update time, and  $A_v$  is the system availability.

The design goal – the objective function of the optimization problem – is to find a system architecture that minimizes the total 10-year lifecycle cost divided by the mean 10-year target probability of detection. The constraint equations represent different requirements the system must satisfy to be considered a viable system architecture. The final six constraint equations place upper and lower bounds on each of the design vector variables.

Now that the DSS conceptual design problem has been mathematically formulated as an optimization problem, it may be solved to find the optimal system architecture(s). As illustrated in Table 2.4, the trade space for a distributed satellite system can be enormous – in this case there exist 732,160 separate system architectures from which to choose. The size of the trade space makes complete enumeration impractical. If it takes approximately 10 minutes of CPU time to simulate and evaluate a single TechSat 21 system architecture, then it would take over 13 years to completely enumerate the trade space! Even if the computation time could be reduced through model reduction or high-speed parallel processing, the memory requirements for storing the parametric data for over half a million different designs could become prohibitive. Thus, appropriate MDO algorithms need to be developed and applied to search the DSS trade space for the best system architectures without completely enumerating the trade space. The process by which this was done and the resulting algorithms used in the MMDOSA methodology are covered in Section 3.5.

Recall that the true mathematical definition of single objective optimization refers to finding the best solution to a problem. However, within the context of exploring the trade space of DSS's, the engineering interpretation of DSS conceptual design optimization is the process of finding good solutions with the intention to find the best solution(s). Table 2.5 lists the number of good and best architectures within the trade space for each of the three DSS conceptual design case studies in Chapters 4-6. Notice how the percentage of "good" and "best" solutions must scale with the magnitude of the trade space to keep the total number of best architectures down to a manageable size.

**TABLE 2.5** Number of "Good" and "Best" Solutions in the Three Case Studies (Single Objective)

Case Study	Trade Space Magnitude	"Good" Architectures	"Best" Architectures
Terrestrial Planet Finder	640	Top 10% 64	Top 1% 6
TechSat 21	732,160	Top 0.1% 732	Top 0.01% 73
Broadband	42,400	Top 1% 424	Top 0.1% 42

---

## 2.4 Summary

This chapter presented definitions for different systems engineering terms within the context of this research, outlined the operations research principles on which the MMDOSA optimization formulation is based, and illustrated how the DSS conceptual design problem may be formally modeled as a mathematical optimization problem. First, the six steps in the space systems design and development process were outlined with emphasis on how important the design decisions made during the Pre-Phase A and Phase A work are in locking in system performance and lifecycle cost. The MMDOSA methodology will provide systems engineers with a tool to aid making such decisions during the conceptual design phase. Next, several common systems engineering terms were explicitly defined, and the associated principles were discussed within the context of the MMDOSA methodology. The structure of a classical optimization problem was then discussed, a classification scheme for different types of optimization problems was presented, solution techniques to such problems were outlined, the differences between single objective and multiple objective optimization problems were pointed out, and the relevance of trade space complexity on the DSS conceptual design problem was analyzed. Finally, the elements of a classical optimization problem were mapped onto the DSS conceptual design problem, and this mapping was used to present the mathematical formulation of an actual distributed satellite system conceptual design problem as an optimization problem. With a solid understanding of systems engineering and operations research principles, and a mathematically sound formulation of the DSS conceptual design problem as an optimization problem, the formal MMDOSA methodology for the conceptual design of distributed satellite systems is now ready to be presented in Chapter 3.



# Chapter 3

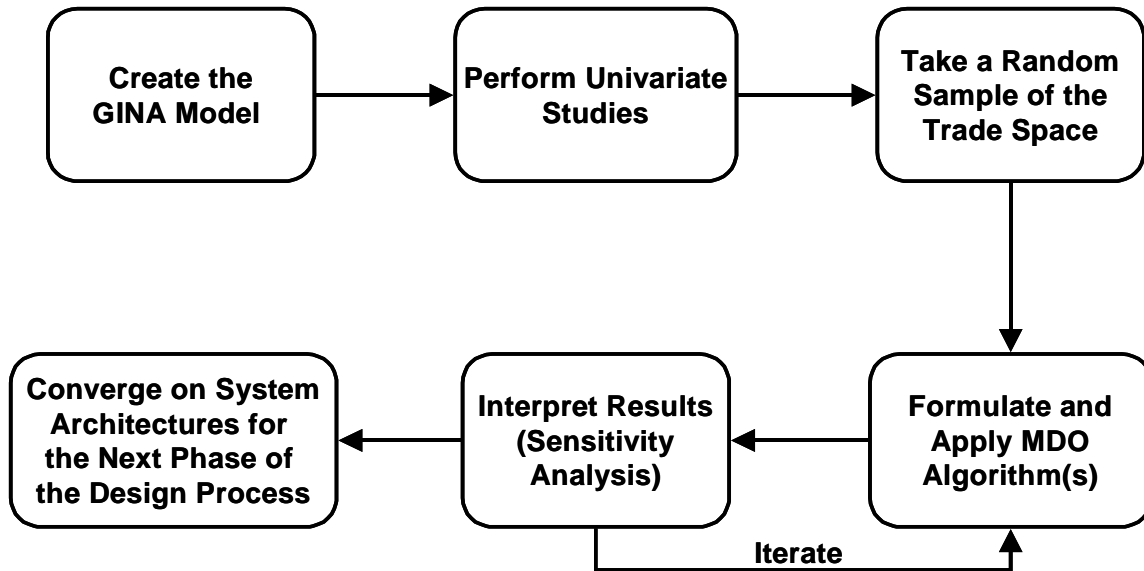
## THE MMDOSA METHODOLOGY

This chapter details the MMDOSA methodology step by step, including the origin of each step, the mathematics and mechanics behind each step, and the utility of each step to the systems engineer in developing a conceptual design for a DSS. Chapter 3 begins with an overview of the MMDOSA methodology and a discussion of how the MMDOSA methodology interacts with the GINA methodology, detailing the similarities and differences between MMDOSA and GINA. Each step of MMDOSA is then presented in sequence, and finally how all the steps fit together to provide a systems engineering tool for the conceptual design of distributed satellite systems is illustrated.

### 3.1 MMDOSA Methodology Overview

Figure 3.1 illustrates the seven steps in the MMDOSA methodology. These steps are:

1. Create the GINA model.
  - Define the mission objective and conceptual design phase objective.
  - Transform the space system into an information network.
  - Develop system metrics.
  - Partition the conceptual design problem.
  - Develop the simulation software.
  - Explore the system trade space.
2. Perform univariate studies.
  - Vary one parameter within a baseline system architecture along a range of values and measure how the system attributes change.
  - Develop an initial feel for the DSS local trade space.



**Figure 3.1** The Multiobjective, Multidisciplinary Design Optimization Systems Architecting Methodology for the Conceptual Design of Distributed Satellite Systems

- Identify model fidelity problems.
3. Take a random sample of the trade space.
    - Obtain a glimpse of the global trade space.
    - Identify initial bounds for each optimization metric.
    - Obtain statistical data to tailor MDO algorithms.
  4. Formulate and apply MDO algorithms.
    - Apply the single objective optimization simulated annealing algorithm to identify the best family(s) of system architectures on the basis of the metric of interest.
    - Apply the multiobjective optimization simulated annealing algorithms to identify the Pareto optimal set of system architectures with respect to all of the decision criteria.
  5. Interpret results (sensitivity analysis).
    - Identify the most influential design variables.
    - Identify the most important high fidelity models.
  6. Iterate.
    - Increase the fidelity of critical models.
    - Modify the simulated annealing algorithm cooling schedule.
    - Change the simulated annealing algorithm DOF parameter.
    - Warm start a new optimization run.
    - Run additional trials of the same optimization algorithm.

7. Converge on system architectures to be focused on in the next phase of the design process.
  - Identify the best family(s) of system architectures from the single objective optimization.
  - Identify the Pareto optimal set of system architectures from the multiobjective optimization.

The remainder of this Chapter details each of these seven steps.

## 3.2 MMDOSA Step 1 – Create the GINA Model

GINA – the Generalized Information Network Analysis methodology for distributed satellite systems – is a systems engineering and architecting (SE&A) framework developed by the MIT Space Systems Laboratory [Shaw, 1998; Shaw et al, 2001]. GINA enables the creation and comparison of different design architectures for a given DSS mission. The fundamental premise behind the GINA methodology is the assertion that most satellite systems are information disseminators that can be represented as information transfer networks [Shaw et al, 2000]. Transforming the system representation from the physical domain to the information network domain enables the systems engineer to compare what physically appear to be very different architectures with the same set of quantitative metrics. Further, this new representation allows the systems engineer to use a body of mathematics that has been developed to analyze data networks to now analyze distributed satellite systems. A summary of the procedural steps in the GINA methodology is listed below.

1. Define the Mission Objective and the Conceptual Design Phase (CDP) Objective.
  - Identify the customer.
  - State mission objective.
  - Derive top-level customer requirements.
  - State CDP objective.
2. Transform the Space System into an Information Network.
  - Identify the origin-destination pairs in the network.
  - Draw the network.
  - Derive the four capability quality of service metrics.
3. Develop System Metrics.

- Define the performance, lifecycle cost, cost per function, and adaptability metrics by which all proposed system architectures will be evaluated and compared.
4. Partition the Conceptual Design Problem.
    - Define the design vector and constants vector.
    - Matrix the design vector against the capability metrics.
    - Identify the modules.
  5. Develop Simulation Software.
    - Develop each module.
    - Code each module.
    - Integrate the coded modules.
  6. Explore the System Trade Space.
    - Evaluate the desired architectures on the basis of the system metrics.

Through these steps, GINA allows the systems engineer to make meaningful, quantitative trades at the conceptual design level by directly relating lifecycle performance to lifecycle cost. The remainder of this section explains each of the above steps in the GINA process and illustrates their application to the Terrestrial Planet Finder (TPF) mission case study. This section is a high level tutorial of the GINA methodology. For a complete discussion of the theory, origin, and application of GINA, please see the following references by Shaw et al [Shaw, 1998; Shaw et al, 2000; Shaw et al, 2001].

### **3.2.1 Relationship Between GINA and MMDOSA**

Figure 3.2 illustrates how the GINA and MMDOSA methodologies interact with each other. The entire GINA methodology may be thought of as the first step of the MMDOSA methodology. This notion of GINA as a precursor and foundation to MMDOSA makes sense as MMDOSA grew out of a critical analysis of the GINA methodology. GINA is a good comparative analysis tool that enables systems engineers to quantitatively analyze and compare existing architectures for a proposed DSS. However, GINA lacks a mechanism to propose and investigate new alternative architectures that may be better than any of the proposed system architectures. This is where MMDOSA comes in – MMDOSA provides a tool to search the DSS trade space for alternative architectures that may optimize the mission. To summarize, both GINA and MMDOSA are systems engineering



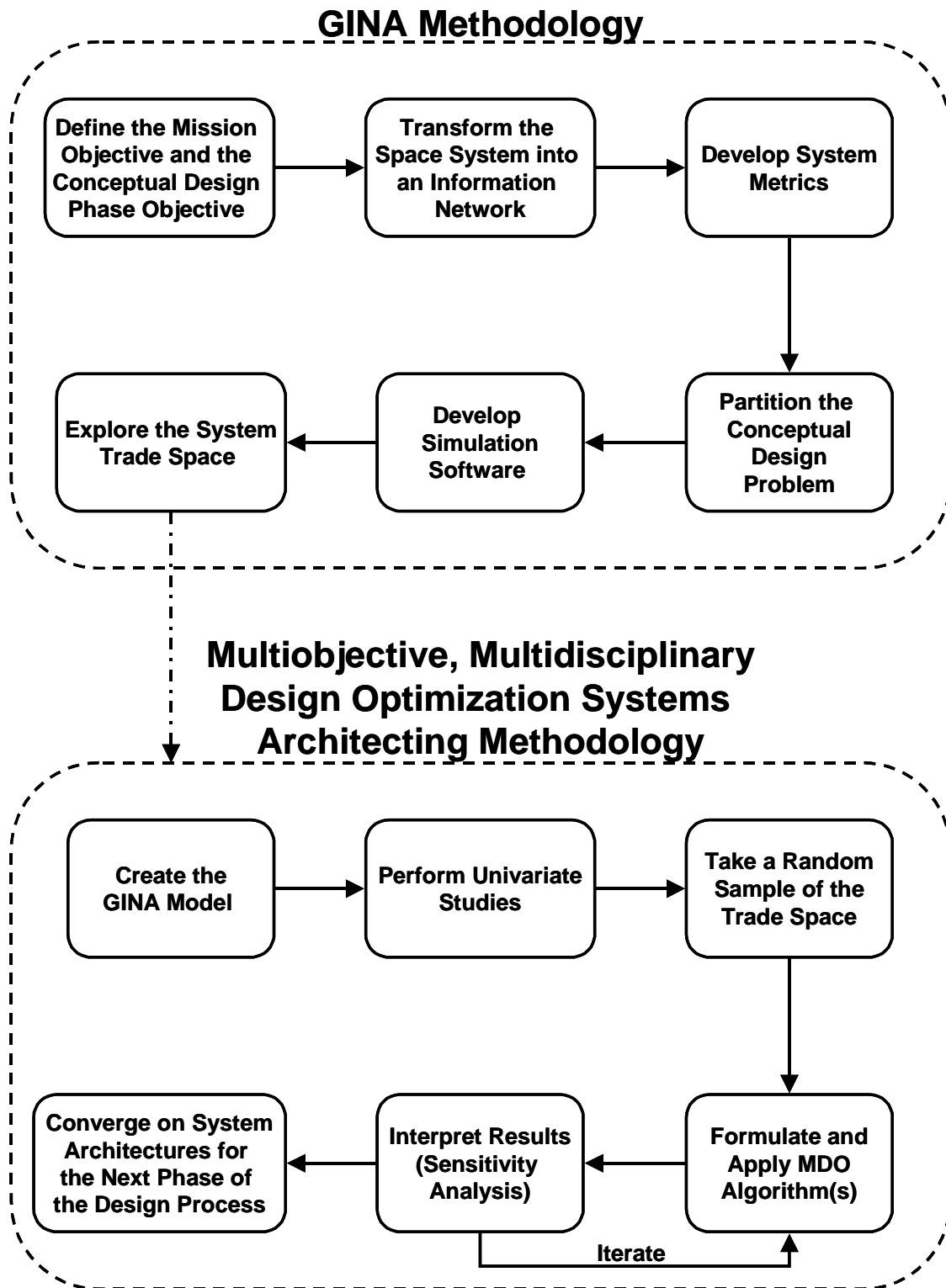


Figure 3.2 Relationship Between GINA and MMDOSA

tools for the conceptual design of distributed satellite systems. However, while GINA is a comparative analysis tool for comparing existing design concepts, MMDOSA is an optimization tool that searches the trade space for the best conceptual designs.

### **3.2.2 GINA Step 1 – Definition of the Mission and CDP Objectives**

In defining the mission objective, the systems engineer must explicitly state what the real application of the system is in terms of the user needs. To do this, the customer of the system must first be identified. For a DSS, the customer is the entity that derives utility from the system, such as troops determining their position from GPS or companies teleconferencing through a broadband satellite network. In the case of TPF mission, there are two customers. The first is the space science and astronomy community. This community will receive all of the scientific data, and would ideally prefer a system that generates the greatest amount of high quality data from as many targets as possible. Because such a system will most likely be financed by a single federal government or through a combination of governments, the second customer is the taxpayer community. In contrast to the scientists, this community would prefer the least expensive system that meets a set of minimum performance requirements. The conflict between these two communities will be captured and illustrated in Chapter 4.

Once the customer has been identified, the mission objective may be defined. For example, the official NASA Origins program mission objective for TPF is:

To study all aspects of planets, ranging from their formation and development in disks of dust and gas around newly forming stars to the presence and features of those planets orbiting the nearest stars. Specifically, to conduct a search for Earth-like planets in star systems located within 15 parsecs of our solar system. – *The TPF Science Working Group* [Beichman et al, 1999]

Close inspection of this statement reveals what the system must accomplish to meet the needs of the user, in this case space scientists. First, the system must detect Earth-like planets in the habitable zone around nearby stars. This will require the use of a Bracewell nulling interferometer. Second, in order to characterize the planets that are discovered to determine whether they contain the elements conducive to life, spectroscopy will be

required to measure the chemical composition of the planets' atmosphere. Thus, defining the mission objective in terms of the user needs begins to elucidate the form and function of the space system to be designed. It also provides the top-level requirements of the system. In the case of TPF, this equates to being able to isolate a planet from its parent star, which may be located up to 15 parsecs from Earth, and being able to actually measure the chemical composition of the planet's atmosphere.

Next, the systems engineer must explicitly define the objective to be achieved during the conceptual design phase to remove all ambiguity concerning what will and will not have been accomplished by the end of this phase of the design process. For the TPF case study, the CDP objective was:

To develop a methodology for the comparison of TPF system architectures spanning from structurally connected to separated spacecraft interferometers. [Miller et al, 2000]

In this example, the goal was not to create a single point design, but rather to create a tool that may be used to model, analyze, and compare different system architectures.

These two objective statements guide the entire GINA process. The mission objective begins the process of identifying key system requirements and how these requirements will be met, while the CDP objective defines the responsibility of the systems engineer during the conceptual design phase in which GINA is being applied. During each of the remaining steps in the application of GINA to the mission at hand, these two statements may be referred back to for confirmation that the work being done is relevant to and aligned with the objectives of the mission and CDP.

### **3.2.3 GINA Step 2 – Information Network Representation**

Modeling a DSS as an information network entails defining all of the origin-destination (O-D) pairs in the system (Figure 3.3). For TPF, the origin is the target star system(s) and the destination is where the data is assembled, either in a computer on the spacecraft itself, at a groundstation, or at a control/data center. Electromagnetic radiation passes through arcs connecting the nodes of this network. Tracking the functionality of this network of

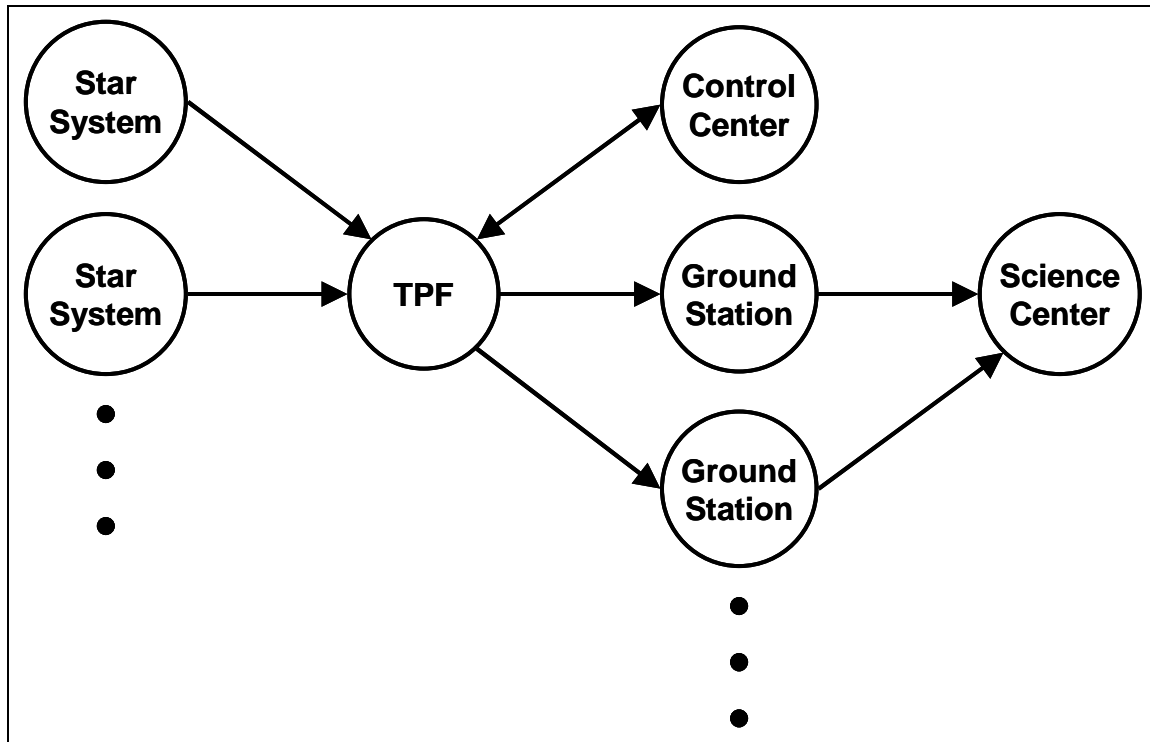


Figure 3.3 The TPF System Modeled as a Network

O-D interconnections, which impacts both the imaging rate and cost, requires probabilistic models of network failures over the lifetime of the mission. A probabilistic reliability model is developed in the performance model in Section 3.2.4. Modeling a space system as an information network also allows for the application of network optimization algorithms to the system design [Ahuja et al, 1993].

In the GINA methodology, the capability of an architecture is characterized by four quality of service parameters that relate to the detection process, and to the quantity, quality, and availability of the information that is processed through the network. These four parameters are signal isolation, information rate, information integrity, and the availability of these services over time [Shaw et al, 2000]. Once formulated, these four parameters serve as the minimum instantaneous capability requirements the system must meet to satisfy the customer. These capability metrics assess how well the customer requirements are met at an instantaneous point in time.

- Isolation

Isolation refers to the ability of a system to isolate and distinguish information signals from different signal and noise sources within the network. For TPF, the system's angular resolution, which is a function of the maximum vector baseline between a pair of collectors, determines the smallest sized objects the interferometer can image and discriminate between in the field of view. Additionally, the null depth and shape of the transmissivity function quantify the ability of the system to null out the parent star and see the planets in the habitable zone. These parameters will be developed in greater detail in the TPF case study in Chapter 4.

- Integrity

Integrity is a measure of the quality of the information being transferred through the network, typically quantified as the bit error rate. In the case of TPF, the integrity of an individual image is a function of the signal-to-noise ratio (SNR) required to obtain that image. The integrity requirements may vary with each mode of operation or each set of measurements in a DSS. Table 3.1 lists the SNR for each mode of TPF operation. TPF architectures with greater integrity will produce images with less ambiguity.

**TABLE 3.1** Integrity Requirements for the Different Detection Modes of TPF

Mode	Science Goal	SNR	Spectral Resolution
Survey	Detect Planet	5	3
Medium Spectroscopy	Detect Atmosphere	10	20
Deep Spectroscopy	Detect Life Markers	25	20

- Rate

Rate measures the speed at which the system transfers information between the nodes (i.e. sources and sinks or O-D pairs) in the network, and is typically measured in bits per second for a network. For TPF, the rate is simply the total number of images the system can produce per unit time and varies for each of the three modes of operation – surveying, imaging, and spectroscopy.

- Availability

Availability characterizes the instantaneous probability that information is being transferred through the network between all of the sources and sinks. TPF targets close to the sun whose imaging needs violate sun avoidance angles within the optical train reduce the availability of the system. The actual imaging time versus the time to complete other tasks such as calibration and retargeting also affects the availability of the system.

The capability quality of service metrics form the underpinning of the remaining GINA analysis. Table 3.2 identifies these metrics for the TPF mission. At this point, the systems engineer must decide which of these capability parameters will become hard requirements the system must meet to be considered feasible and which parameters will serve to distinguish between architectures. For TPF, the isolation, integrity, and availability parameters are hard requirements every potential design must meet, while the imaging rate will differ between architectures. Imaging rate is the inverse of the sum of the integration time and overhead time involved in obtaining an image, and is different for each of the three modes of operation.

**TABLE 3.2** The GINA Capability Quality of Service Metrics For TPF

<b>Capability</b>	<b>TPF Equivalent</b>
Isolation	Angular Resolution, Null Depth
Integrity	Images Per Unit Time
Rate	SNR
Availability	% Time in Use

### 3.2.4 GINA Step 3 – Develop System Metrics

To compare all the different architectures formulated during the conceptual design phase, GINA uses a quantifiable set of metrics – capability, performance, lifecycle cost, cost per function, and adaptability. The capability metrics have already been introduced in Section 3.2.3. The remaining four metrics are discussed below. These are the metrics that will be used to evaluate and compare all of the DSS architectures – in this case all of the TPF architectures.

#### **Performance**

While the four capability quality of service metrics measure how well the architecture meets the capability requirements at any instantaneous point in time, the performance metric measures how well the architecture satisfies the demands of the market over the entire life of the mission [Shaw, 1998]. Because imaging rate was identified as the distinguish-

ing capability metric between TPF architectures, performance is expressed as the total number of images the system produces over the mission duration. In this case, "images" refers to the sum of all surveys, medium spectroscopy's, and deep spectroscopy's. To calculate this quantity, however, the additional complication of taking into account all the possible failures that may occur within the system must be addressed. As individual components within the system fail over time, the imaging rate of the system will decrease in order to maintain the same pre-failure level of integrity (i.e. TPF must still achieve the same SNR for each operational mode, despite failures). Thus, a reliability model is needed to take into account potential failures and the effect they have on system performance.

GINA uses Markov reliability modeling techniques to determine both the probability that the system will continue to function over a given amount of time and the likelihood with which the system will function in different partially failed states throughout the mission. A Markov model must be created for each proposed architecture. From the Markov model, a system of differential equations is developed, as shown in Eqn. 3.1, where  $\dot{\vec{P}}$  is the time rate of change of the state probability vector,  $A$  is the state coefficient matrix comprised of component failure rates, and  $\vec{P}$  is the state probability vector.

$$\dot{\vec{P}} = A\vec{P} \quad (3.1)$$

The solution to this set of linear, first-order, partially coupled ordinary differential equations determines the probability of the DSS being in any given state at a particular time. Coupling the outputs of the reliability model with the outputs of the capability model (i.e. in this case the imaging rate of the TPF architecture in each operational state) yields the total performance (i.e. total number of images) of the system. The coupling equation is called the utility function. A generalized utility function that may be applied to compute the performance of any DSS is given in Eqn. 3.2.

$$U(T) = \int_0^T \sum_{i=1}^n C_i P_i(t) dt \quad (3.2)$$

$U$  is the total utility or performance (i.e. number of images for TPF),  $T$  is the mission lifetime,  $n$  is the total number of operational states,  $C_i$  is the capability in each state  $i$  (i.e. imaging rate for TPF), and  $P_i(t)$  is the probability of being in each operational state  $i$  as a function of time  $t$ .

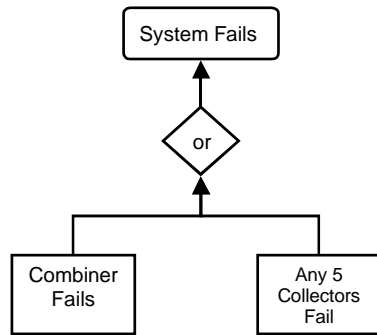
An example of how to create a Markov model for a DSS architecture will now be presented within the context of the TPF analysis – in this case an eight-collector nulling interferometer. Markov models have traditionally been used to evaluate the reliability of complex systems. To apply a Markov modeling methodology, the states of the system must be time dependent, sequential, and mutually exclusive. If the system satisfies these requirements, a set of differential equations can be written to model the evolution of the system. By taking advantage of the Markov property, which states that given full knowledge of the current state of the system, one can predict all future states by integrating the set of differential equations, irrespective of the past states of the system; the probability of being in any given state of the system at any given time may be computed [Gelb, 1974].

The first step is to develop a fault tree diagram illustrating all the possible modes of failure for an architecture (Figure 3.4). The minimum functionality required for a nulling interferometer is one combiner and four collector apertures. Thus, the eight-collector TPF architecture fails when the combiner fails or when any five collectors fail. The system will still function when one to four collector apertures fail, but at a reduced capability (i.e. reduced imaging rate for TPF).

From the fault tree, a Markov model illustrating each possible state of the system may be created for each architecture. Figure 3.5 illustrates the Markov model for the eight-collector TPF architecture. This model contains five possible operational states, all of which require a functional combiner.

- State 1: All eight collectors are working.
- State 2: Seven of the eight collectors are working.
- State 3: Six of the eight collectors are working.

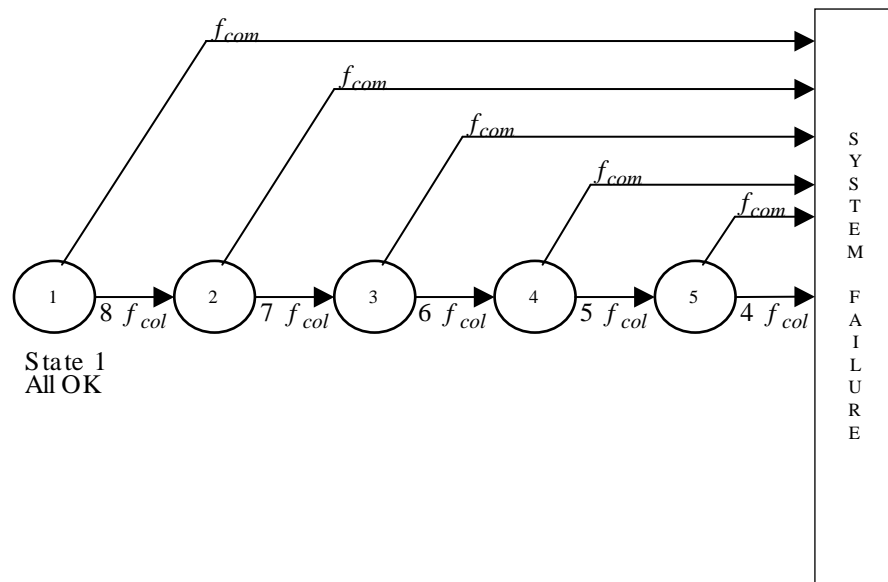




**Figure 3.4** Aggregated Fault Tree for an Eight-Collector TPF Architecture

- State 4: Five of the eight collectors are working.
- State 5: Four of the eight collectors are working.

Otherwise, the system is in a state of failure because the nulling (i.e. isolation) requirement can no longer be met.



**Figure 3.5** Markov Model State Diagram for the Eight-Collector TPF Architecture

From the Markov model state diagram, a system of differential equations can be written to determine the probability of the system being in any given state at any given time. This is done by representing each possible state of the system in the Markov model as a node in a

network. To determine the differential equation for a particular state, the flow into and out of the node representing that state is balanced. The eight-collector TPF architecture requires a set of five partially coupled, linear, first-order differential equations to model the system (Eqn. 3.3).

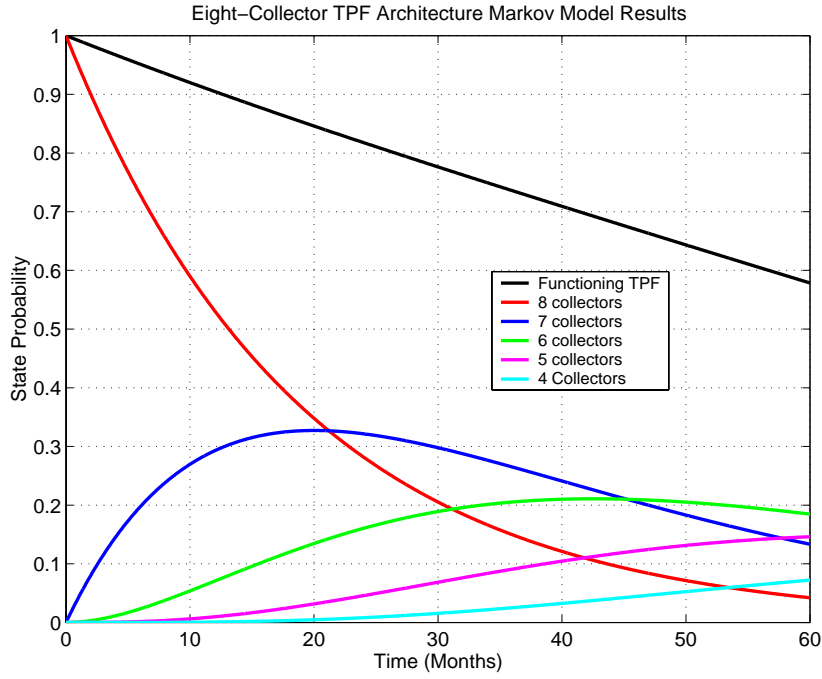
$$\begin{bmatrix} \dot{P}_1 \\ \dot{P}_2 \\ \dot{P}_3 \\ \dot{P}_4 \\ \dot{P}_5 \end{bmatrix} = \begin{bmatrix} -(f_{com} + 8f_{col}) & 0 & 0 & 0 & 0 \\ 8f_{col} & -(f_{com} + 7f_{col}) & 0 & 0 & 0 \\ 0 & 7f_{col} & -(f_{com} + 6f_{col}) & 0 & 0 \\ 0 & 0 & 6f_{col} & -(f_{com} + 5f_{col}) & 0 \\ 0 & 0 & 0 & 5f_{col} & -(f_{com} + 4f_{col}) \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \end{bmatrix} \quad (3.3)$$

Knowledge of the failure rates for the combiner ( $f_{com}$ ) and collector ( $f_{col}$ ) are required to solve for the state probabilities. The failure rate ( $f$ ) is the inverse of the *mttf* (Eqn. 3.4).

$$f = \frac{1}{MTTF} \quad (3.4)$$

The system's initial conditions are also required for the solution. In this case, the initial conditions (i.e. at  $t=0$  in the beginning of the operational mission) are a 100% probability of being in state 1 and a 0% probability of being in all successive states. The plot in Figure 3.6 illustrates the results for the eight collector TPF example. The Markov model gives the exact probability of being in any of the five operational states as a function of time through the five-year TPF mission. Each of these five operational states will have a different survey, medium spectroscopy, and deep spectroscopy imaging rate. The markov models for all other TPF, TechSat 21, and broadband satellite architectures were implemented and solved in the same manner, and this approach is generalizable to any DSS.

The solution to this system of differential equations (Eqn. 3.3) determines the probability  $P_i(t)$  of the system being in any given state  $i$  at a particular time  $t$ . Coupling, via the utility function in Eqn. 3.2, the outputs of the reliability model with the outputs of the capability model yields the total performance (i.e. total number of images  $\Psi$ ) of the system. The coupling equation for the TPF case study is:



**Figure 3.6** Markov Model Results for the Eight-Collector TPF Architecture ( $f_{com}=5.5 \times 10^{-3} \text{ month}^{-1}$ ,  $f_{col}=8.3 \times 10^{-3} \text{ month}^{-1}$ )

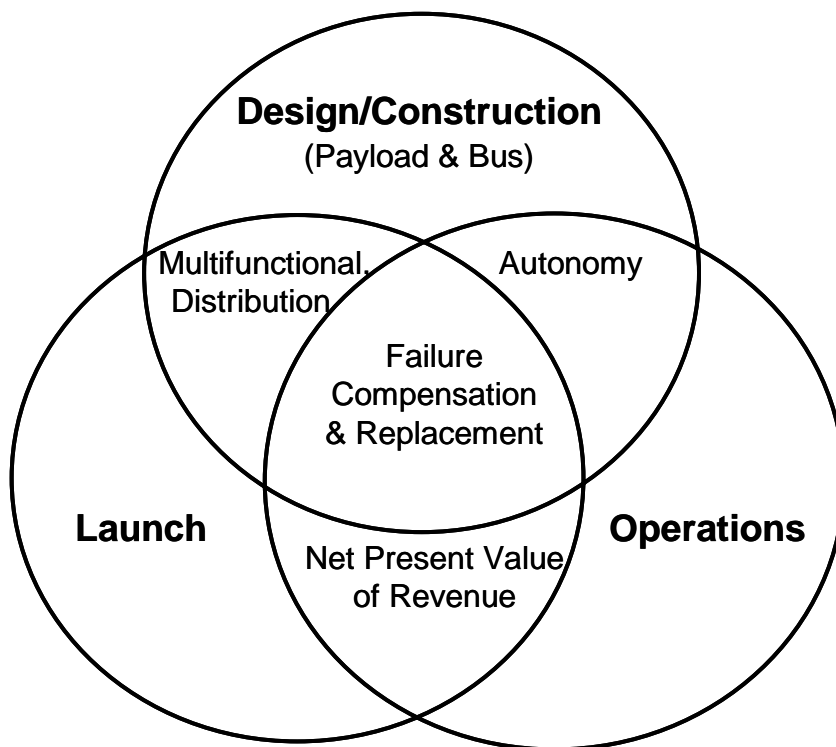
$$\begin{aligned}
 \sum_{y=1}^5 \Psi_y &= \int_{74}^{256} \sum_{i=1}^n I_{si} P_i(t) dt + \int_{257}^{313} \sum_{i=1}^n I_{mi} P_i(t) dt + \int_{314}^{365} \sum_{i=1}^n I_{di} P_i(t) dt \\
 &+ \int_{366}^{547} \sum_{i=1}^n I_{si} P_i(t) dt + \int_{548}^{658} \sum_{i=1}^n I_{mi} P_i(t) dt + \int_{659}^{730} \sum_{i=1}^n I_{di} P_i(t) dt \\
 &+ \int_{731}^{804} \sum_{i=1}^n I_{si} P_i(t) dt + \int_{805}^{1004} \sum_{i=1}^n I_{mi} P_i(t) dt + \int_{1005}^{1095} \sum_{i=1}^n I_{di} P_i(t) dt \\
 &+ \int_{1096}^{1132} \sum_{i=1}^n I_{si} P_i(t) dt + \int_{1133}^{1333} \sum_{i=1}^n I_{mi} P_i(t) dt + \int_{1334}^{1460} \sum_{i=1}^n I_{di} P_i(t) dt \\
 &+ \int_{1461}^{1498} \sum_{i=1}^n I_{si} P_i(t) dt + \int_{1499}^{1680} \sum_{i=1}^n I_{mi} P_i(t) dt + \int_{1681}^{1825} \sum_{i=1}^n I_{di} P_i(t) dt
 \end{aligned} \tag{3.5}$$

where  $y$  is the year in the mission, the limits of integration denote time steps of one day,  $i$  is an index indicating a particular operational state,  $n$  is the total number of operational states ( $n=5$  for the eight-collector aperture TPF example),  $I_s$  is the survey mode imaging rate,  $I_m$  is the medium spectroscopy mode imaging rate,  $I_d$  is the deep spectroscopy mode imaging rate, and  $P_i(t)$  is the probability of being in any state  $i$  as a function of time  $t$  from Eqn. 3.3. As one can see, the actual implementation of the generalized utility function (Eqn. 3.2) can be quite involved and complex for a real DSS with multiple modes of operation. The total system performance is obtained by summing fifteen separate utility functions, where each row in Eqn. 3.5 denotes a separate year in the five-year mission and each column denotes a different mission mode (i.e. surveying, medium spectroscopy, and deep spectroscopy).

### **Lifecycle Cost**

The Venn diagram in Figure 3.7 illustrates all of the costs generally taken into account when determining the lifecycle cost of a DSS. The three primary cost categories are the development of the spacecraft payloads and buses, deployment (i.e. launch) of the system, and the operation of the system. Secondary cost trades appear within the intersections of these primary categories. For example, adding autonomy to the system lowers operations costs, but increases design costs. Distributing payloads on multiple, identical satellites reduces design costs at the expense of launch costs, while increasing the multifunctionality of the satellites does the opposite. Furthermore, distribution reduces failure compensation costs by allowing for the replacement of only the components within the system that have failed. Finally, spending more money for early deployment at the beginning of a commercial program to create an initial revenue stream can increase the total net present value and internal rate of return on the project due to the time value of money.

The spacecraft bus cost model estimates the total cost involved in designing, manufacturing, integrating, and testing each satellite bus. This cost will generally be proportional to the mass, power, and other subsystem parameters of the spacecraft. Over the years, several governmental organizations have created Cost Estimation Relationships (CERs) for



**Figure 3.7** DSS Lifecycle Costs

satellite buses that relate how the cost properties of a spacecraft bus vary with the subsystem parameters of the bus. These CERs are based on historical data of past satellite programs, and work on the assumption that future costs will reflect historical trends. This is a tenuous assumption given the rapid evolution of space technology, but represents the current state-of-the-art in satellite bus cost modelling. The two most commonly used CERs are the U.S. Air Force's Unmanned Spacecraft Cost Model (USCM) (Table 3.3) and the Aerospace Corporation's Small Satellite Cost Model (SSCM) (Table 3.4 and Table 3.5). The SSCM is valid for satellites of approximately 500 kg or less and is based on 1990's technology, while the USCM database contains much larger satellites, but is based on older 1970's and 1980's technology.

While any of these CERs may be used separately to obtain an independent estimate of the theoretical first unit (TFU) satellite bus cost, it is generally more effective to use several CERs simultaneously to reduce the effect of statistical outliers in the historical

TABLE 3.3 Air Force USCM CERs [Larson &amp; Wertz, 1992]

CER Cost Driver	Parameter $x$ (units)	Applicable Range	CER (FY92\$K)	Standard Error (FY92\$K)
IR Payload	Aperture Diameter (m)	0.2-1.2	$\$=122,758x^{0.562}$	18,425
Visible Payload	Aperture Diameter (m)	0.2-1.2	$\$=44,263x^{0.562}$	6651
Communication Antenna	Mass (kg)	1-87	$\$=20+230x^{0.59}$	476
Communication Electronics	Mass (kg)	13-156	$\$=179x$	8235
Spacecraft Bus	Dry Mass (kg)	26-1237	$\$=185x^{0.77}$	6655
Structure/Thermal	Mass (kg)	7-777	$\$=86x^{0.65}$	1247
TT&C	Mass (kg)	4-112	$\$=93+164x^{0.93}$	1565
Attitude Determination	Dry Mass (kg)	6-97	$\$=1244x^{0.39}$	1912
Attitude & Reaction Control	Dry Mass (kg)	9-167	$\$=-364+186x^{0.73}$	999
Power	EPS Mass. x BOL Power (kg-W)	104-414, 920	$\$=183x^{0.29}$	2254

database of satellites. A weighted average algorithm is employed using the standard error of each CER to establish the appropriate weighting factor to estimate the satellite bus cost from multiple CERs [Wertz & Larson, 1996].

$$Cost_{s/c\ bus} = \frac{\sum \frac{\$_i}{\sigma_i^2}}{\sum \frac{1}{\sigma_i^2}} \quad (3.6)$$

where  $Cost_{s/c\ bus}$  is the TFU cost of the spacecraft bus,  $\$_i$  is the  $i^{th}$  cost estimate, and  $\sigma_i$  is the  $i^{th}$  standard error.

CERs predict the TFU cost of a satellite bus, including all of the design work that takes place before the actual construction of the bus. However, CERs poorly estimate the costs for even the most modest production lines. To take into account the fact that multiple payloads and satellite buses will be built for a DSS, a learning curve will be applied to the TFU bus costs and TFU payload costs estimated from the CERs [Larson & Wertz, 1992]. Using the learning curve, the total production cost for  $N$  units is:

**TABLE 3.4** The Aerospace Corporation SSCM Version 7.4 CERs [Wertz & Larson, 1996]

Independent Variable	# Data Points	CER for Total Bus Cost (FY94\$M)	Applicable Range	Standard Error (FY94\$M)
Satellite volume (in <sup>3</sup> )	12	$\$=-34.84+4.66\ln(x)$	2,000-80,000	4.27
Satellite bus dry mass (kg)	20	$\$=0.704+0.0235x^{1.261}$	20-400	3.33
ACS dry mass (kg)	14	$\$=6.65+0.042x^2$	1-25	5.45
TT&C subsystem mass (kg)	13	$\$=2.55+0.29x^{1.35}$	3-30	4.50
Power system mass (kg)	14	$\$=-3.58+1.53x^{0.702}$	7-70	3.52
Thermal control mass (kg)	9	$\$=11.06+0.19x^2$	5-12	5.37
Structures mass (kg)	14	$\$=1.47+0.07x\ln(x)$	5-100	5.40
Number of thrusters	5	$\$=46.16-41.86x^{-0.5}$	1-8	8.95
Pointing accuracy (deg)	16	$\$=1.67+12.98x^{-0.5}$	0.25-12	7.37
Pointing knowledge (deg)	10	$\$=12.94-6.681x\ln(x)$	0.1-3	8.79
BOL power (W)	16	$\$=-22.62+17.9x^{0.15}$	20-480	6.13
Average power (W)	17	$\$=-8.23+8.14x^{0.22}$	5-10	5.71
EOL power (W)	14	$\$=0.507+1.55x^{0.452}$	5-440	6.20
Solar array area (m <sup>2</sup> )	13	$\$=-814.5+825.7x^{0.0066}$	0.3-11	6.37
Battery capacity (A-hr)	12	$\$=1.45+1.91x^{0.754}$	5-32	6.01
Data storage cap (MB)	14	$\$=-143.5+154.85x^{0.0079}$	0.02-100	8.46
Downlink data rate (kbps)	18	$\$=26.0-21.86x^{-0.23}$	1-1000	8.91

**TABLE 3.5** The Aerospace Corporation SSCM Version 8.0 CERs [Wertz & Larson, 1996]

Independent Variable	# Data Points	CER for Total Bus Cost (FY94\$M)	Applicable Range	Standard Error (%)
r: EOL power (W)	17	$\$=6.47r^{0.1599}s^{-0.356}$	r: 5-500	29.55
s: Pointing accuracy (°)			s: 0.05-5	
r: TT&C mass (kg)	18	$\$=0.702r^{0.554}s^{0.0363}$	r: 3-50	35.68
s: Payload power (W)			s: 10-120	
r: Downlink data rate (kbps)	21	$\$=1.44r^{0.0107}s^{0.509}1.0096^p$	r: 1-2000	35.66
s: Average power (W)			s: 5-410	
p: Prop system dry mass (kg)			p:-35	
r: Spacecraft dry mass (kg)	26	$\$=0.6416r^{0.661}-1.5117s^{0.289}$	r: 20-400	37.19
s: Pointing accuracy (deg)			s: 0.05-5	
r: Solar array area (m <sup>2</sup> )	20	$\$=4.291r^{0.255}1.989^s$	r: 0.3-11	38.53
s: ACS type (3-axis or other)			s: 0=other, 1=3-axis	
r: Power subsystem mass (kg)	25	$\$=0.602r^{0.839}$	r: 7-70	37.07

$$B = 1 - \frac{\ln\left(\frac{100\%}{S}\right)}{\ln 2} \quad (3.7)$$

$$L = N^B$$

$$Cost_{production} = TFU \times L$$

where  $L$  is the learning curve factor and  $S$  is the learning curve slope in percent. In the aerospace industry,  $S$  varies between 85%-95% [Larson & Wertz, 1992]. Table 3.6 lists which values of  $S$  that Robert Wong, Manager of Economic Analysis in the Systems Engineering Division of TRW, recommends as a function of production quantity [Larson & Wertz, 1992].

**TABLE 3.6** Learning Curve Slopes In the Space Industry

Production Quantity	Recommended S
<10	95%
10-50	90%
>50	85%

Launch costs are estimated from published look up tables for the selected launch vehicles. The *International Guide to Space Launch Systems* [Isakowitz, 1995] is a good reference for launch costs. Appendix B outlines the optimization approach used in the GINA model launch module of the three DSS case studies in Chapters 4-6 to solve the launch vehicle selection problem. The selection problem is based on finding the optimal subset of launch vehicles that can deploy all of the satellites in a constellation at the minimum cost and/or risk; while at the same time adhering to a set of satellite, orbital dynamics, political, and availability constraints. The optimization approach developed in Appendix B determines the optimal subset of launch vehicles. The core of the approach is a database that contains information on all of the operational launch vehicles in the market. Information from this database is then combined with the properties of the satellite cluster or constellation to be deployed to create a mathematical formulation of the launch vehicle selection problem as an integer program (IP). The resulting IP is then solved via a branch-and-bound algo-



rithm. In this manner, optimal solutions are found for different DSS deployment, replacement, and replenishment scenarios. Finally, operations costs may be estimated from the control center crew size, the amount of software needed, and the number of ground stations used.

In contrast to the lifecycle cost models for civil and military missions, the lifecycle cost models for commercial missions must also take into account the revenues produced by the DSS. In these cases, parameters such as net present value (NPV), capital recovery, and the present value of profit (PVP) used by the financial community to assess the economic competitiveness of new business ventures can be incorporated into the DSS lifecycle cost model. The present value  $P$  of a future sum  $F$  at  $Y$  years may be computed as:

$$P = F(1 + r)^{-Y} \quad (3.8)$$

where  $r$  is the discount rate. The capital recovery formula defines the stream of  $N$  constant payments  $R$  that are equivalent to the present sum  $P$  [de Neufville, 1990], and may be calculated as:

$$R = \frac{P[r(1 + r)^N]}{[(1 + r)^N - 1]} \quad (3.9)$$

Present value of profit  $PVP$  for a commercial mission is:

$$PVP = R(1 + r)^{-Y} - LCC(1 + i)^Y(1 + r)^{-Y} \quad (3.10)$$

where  $R$  is revenue,  $LCC$  is the lifecycle cost, and  $i$  is the interest rate. These revenue-related components of the lifecycle cost model are used in the broadband communication case study in Chapter 6.

### **Cost Per Function**

The cost per function (CPF) metric provides a clear measure of the cost of an architecture versus its performance, and is computed by dividing the lifecycle cost of a DSS architecture by its lifecycle performance. Examples include the cost per billable minute of a tele-

communications system or the cost per megabyte of data for a weather reconnaissance system. Table 3.7 lists some other examples of the CPF metric. For TPF, the cost per function is defined as the cost per image, and is calculated by dividing the total lifecycle cost of TPF by the total number of images it produces over its mission life. The performance of the system is computed by Eqn. 3.2 in general and by Eqn. 3.5 for the TPF example.

**TABLE 3.7** The Cost Per Function Metric for Different DSS Missions [Shaw et al, 2001]

<b>DSS</b>	<b>Cost Per</b>	<b>Satisfied</b>	<b>User</b>
Mobile Communication	Cost Per	Billable	Voice-Circuit Minute
Broadband Communication	Cost Per	Billable	T1-Minute
GMTI Radar	Cost Per	Protected	km <sup>2</sup> of Theater
Weather Reconnaissance	Cost Per	Useful	Megabyte
Astronomical Telescope	Cost Per	Useful	Image

### **Adaptability**

In GINA, adaptability is a measure of how flexible an architecture is to changes in design assumptions and mission requirements [Saleh, 2002]. In one sense, adaptability may be thought of as the sensitivity or elasticity of the CPF of a particular architecture to incremental changes in an assumption or requirement. For the TPF mission, potential assumptions that could be altered to measure architecture sensitivity include component costs, learning curve slopes, and component reliabilities. In another sense, adaptability may be thought of as the flexibility of a particular architecture to a new set of mission requirements. An example of flexibility for TPF might be the ability of an architecture to transition from a planetary detection mission to an astrophysical imaging mission.

### **3.2.5 GINA Step 4 – Partition the Conceptual Design Problem**

Next, all the design variables the systems engineer has control over are listed. From this list, key independent variables that drive the design of the system are extracted and inserted into a design vector. The remaining variables are placed in the constants vector.

For TPF, four variables – heliocentric orbital radius, collector connectivity/geometry, number of collector apertures, and collector aperture diameter – make up the design vector. Table 3.8 lists the range of discrete values considered for each design vector variable in the TPF case study. Each set of values in a design vector defines a unique TPF system architecture.

**TABLE 3.8** The TPF Design Vector

Variable	Value Range
Heliocentric Orbital Radius (AU)	1-5.5
Collector Connectivity/Geometry*	SCI-1D, SCI-2D, SSI-1D, SSI-2D
# Collector Apertures	4, 6, 8, 10, 12
Collector Aperture Diameter (m)	1 m - 4 m

\*SCI, structurally connected interferometer; SSI, separated spacecraft interferometer; 1D, one-dimensional array; 2D, two-dimensional array

Once the design vector has been identified, it may be matrixed against the four capability quality of service metrics and the system metrics. The chart in Figure 3.8 is a cornerstone of this step as it establishes the relationship between the trade space for the DSS and the metrics by which competing architectures are judged. Notice the fundamental relationships between the elements of the design vector and the capability metrics. For example, the number of apertures in the system will directly affect the ability to shape the transmissivity function. Hence, the number of apertures drives the isolation metric (angular resolution). The different attributes can be lumped into groups of modeling needs that allow the recognition of important differences between competing architectures. These groups directly determine the modules that are required to capture the TPF-relevant relationships of physics, cost, and systems engineering trades [Miller et al, 2000].

The systems engineer may now identify the modules that will be required to develop, analyze, and compare different architectures for the DSS being designed. These modules capture the fundamental physics of the problem. For TPF, six core modules are required:

1. Environment module

<b>Top Trades Capability Metrics</b>	Heliocentric Orbital Radius (1 to 6 AU)	Aperture Maintenance (SCI vs. SSI)	Number of Apertures (4 to 12)	Size of Apertures (1 to 4 m)
Isolation (Angular Res.)	N/A	SSI allow more freedom in baseline tuning	Fine tuning of transmissivity function	N/A
Rate (Images/Life)	Noise reductions increase rates. Different operation delays	SSI power and propulsion requirements highly sensitive	Increased collecting area improves rate	Increased collecting area improves rate
Integrity (SNR)	Different local zodiacal emission and solar thermal flux	SCI: passive alignment but complex flexible dynamics	Tuning of transmissivity for exo-zodiacal suppression	Smaller FOV collects less local zodiacal noise
Availability (Variability)	N/A	Different safing complexity and operational events	Different calibration and capture complexity	N/A

Aperture

Systems Analysis

Operations

Dyn & Controls

Environment

S/C

**Figure 3.8** Terrestrial Planet Finder Model Component Identification Metrics Matrix [Curtis et al, 1999]

2. Aperture Configuration module
3. Spacecraft Payload and Bus module
4. Dynamics, Control, & Stability module
5. Deployment & Operations module
6. Systems Analysis module

### 3.2.6 GINA Step 5 – Develop Simulation Software

A module is a Matlab m-file that models a specific subsystem, component, or function within the mission. A module within the simulation software may be defined as a piece of computer code which performs an independent set of calculations, contains a single entry and exit point, and may be tested in isolation [Kockler, F.R., 1990]. Attributes of a good modular unit within the simulation software include high internal coupling within the module, lower coupling between modules, and the minimization of feedback loops incor-

porating two or more modules. The purpose of the modules is to develop a systematic tool that sizes the components within an architecture for a given set of inputs. If implemented correctly, the modules should also aid in the systems analysis by computing the system metrics for a given architecture. This allows for rapid evaluations and comparisons of alternative architectures. The steps in the simulation software development process (Figure 3.9) are discussed below.

First, the simulation software objectives and requirements are defined in accordance with the mission and CDP objectives formulated in Section 3.2.2. Next, the software modules required to model the system are identified via the model component identification matrix (Figure 3.8). These modules may be divided into submodules. For example, the spacecraft bus module may be divided into power, propulsion, etc. submodules. Once all of the modules and submodules have been identified, interface control becomes very important. This includes explicitly defining the inputs required by the module, outputs the module will deliver, and mathematical relationships between these inputs and outputs.

Next, each module may be coded concurrently and used to explore the trade space of a portion of the design architecture (ex. How does solar array area scale with orbital radius?). By understanding the local trade space of different portions of the system, the systems engineer will be better prepared to make intelligent design decisions and to interpret the results of any attempts to optimize a design. This step also provides an opportunity to debug the module before integrating it with the other modules.

Once a module has been debugged, it may be coded up as a Matlab function. A master m-file then links and integrates the modules together. At this point, interface control becomes an important issue. An  $N^2$  diagram may be used as the interface control mechanism during this step. Figure 3.10 illustrates the  $N^2$  diagram for the TPF GINA model. As one can see, certain modules were further subdivided into submodules. This modular division of the TPF architecture reduces software development risk by reducing coupling and simplifies the simulation code development as each module and submodule is sepa-

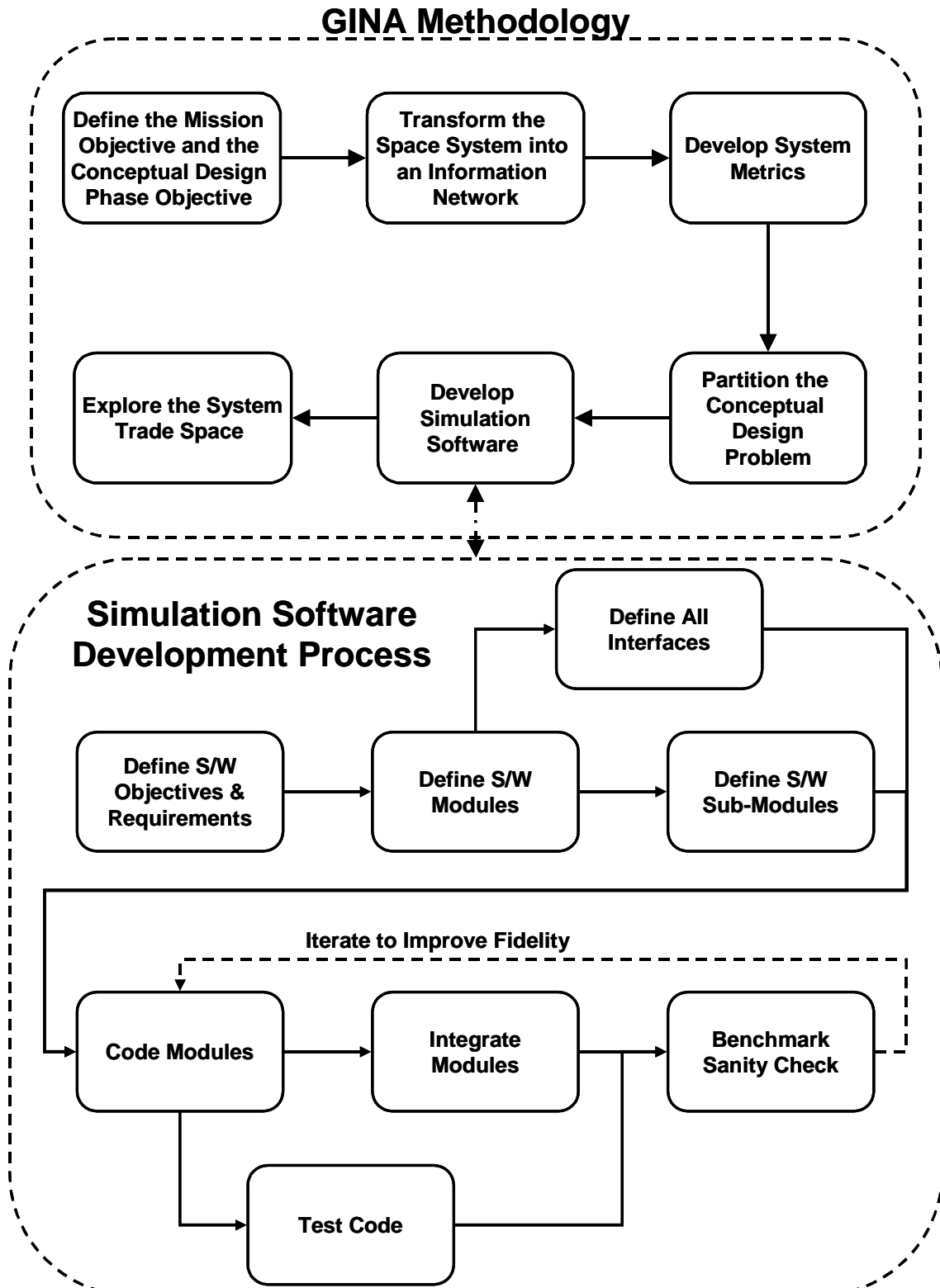


Figure 3.9 Relationship Between GINA and the Simulation Software Development Process

rately testable. An  $N^2$  diagram, sometimes called a design structure matrix (Browning, 1998), is an  $N \times N$  matrix used by systems engineers to develop and organize interface information [Kockler, F.R., 1990]. The submodules (i.e. Matlab m-file functions) are located along the diagonal of the matrix. Inputs to each submodule are vertical and the outputs are horizontal. The aggregation of the submodules into modules is illustrated by the black boxes enveloping different sections of the diagonal. The  $N^2$  diagram in Figure 3.10 provides a visual representation of the flow of information through the simulation software and is used to connect all of the Matlab functions to enable an automated simulation of different TPF architectures.

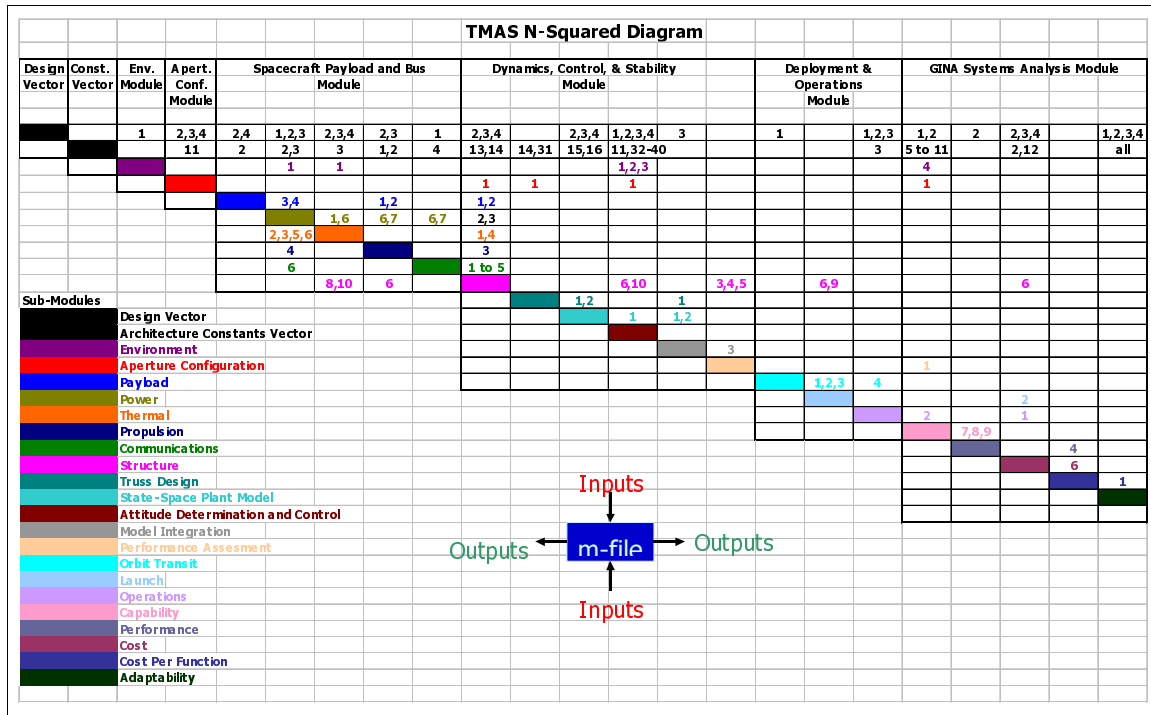


Figure 3.10  $N^2$  Diagram for the TPF GINA Model

Outputs from each module serve as inputs to one or more successive modules to size the architecture and evaluate it on the basis of the system metrics. At this point, a simulation tool has been created. For any given set of inputs (i.e. design vector and constants vector), the Matlab master m-file will use the modules to size and evaluate that architecture. Results from different simulations may then be used to compare different architectures.

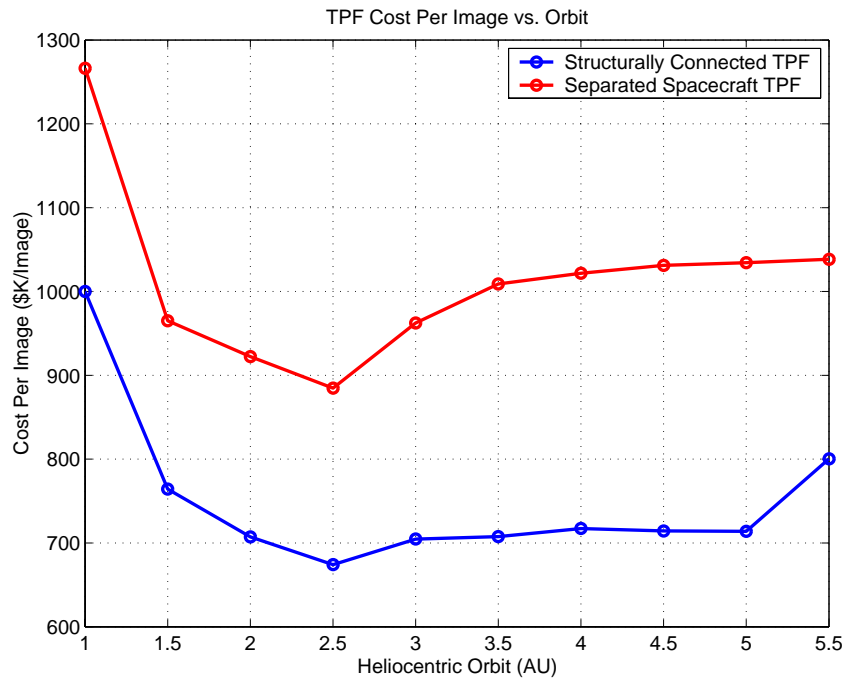
Before the systems engineer can use the GINA model to explore the trade space and compare different DSS architectures, however, the validity of the model must be ascertained. A series of test cases, or benchmarks, need to be executed to ensure that the results from the simulation (i.e. the modules) are reasonable. This provides a sanity check and ensures that the systems engineer is not simply using a "garbage in-garbage out" model to explore the trade space. For the TPF case study, the results from the TPF GINA model were compared to the TPF system parameters for different industry point designs to see which estimates had significant differences, and to explain the reasons behind these differences. It is important to point out that the goal in benchmarking is not to match exactly the results from the existing industry point designs, but rather to establish a coarse correlation – taking into account the different assumptions that went into the GINA model versus the industry models – and to use this information to improve the fidelity of the modules.

### **3.2.7 GINA Step 6 – Explore the System Trade Space**

Once the simulation software has been fully integrated and benchmarked, different system architectures may be tested and compared on the basis of the CPF metric or other metrics. The selection of the design vector is at the discretion of the systems engineer. One possible strategy is to choose a baseline design vector and then vary a single variable across its entire range of values. Figure 3.11 illustrates how the cost per image varies as a function of orbit for linear structurally connected and separated spacecraft TPF baseline architectures with four two-meter collector apertures.

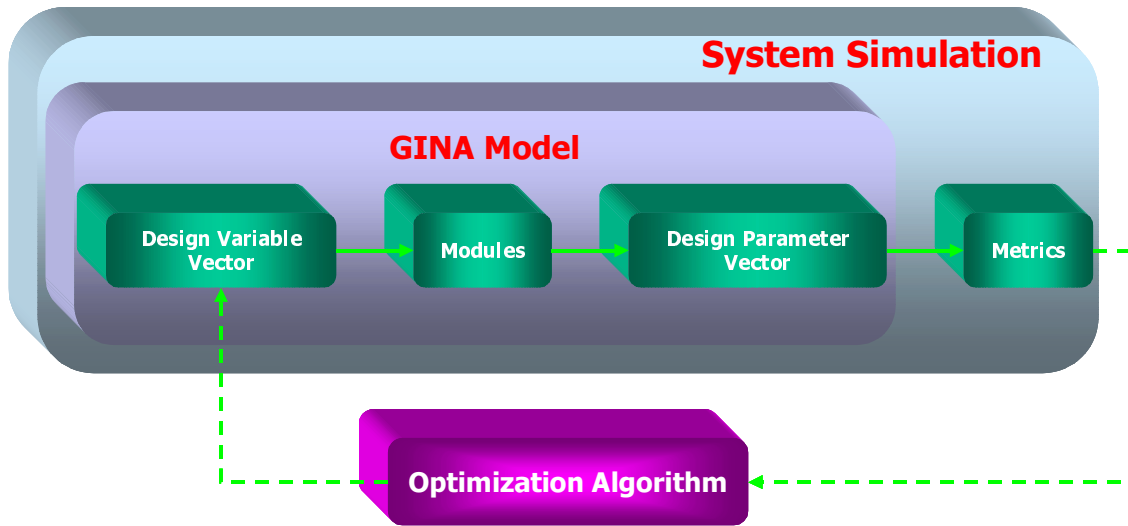
Varying only a single parameter at a time, however, ignores many of the nonlinear couplings between various system parameters. Rather, the systems engineer would like to know how the system varies when all of the design parameters vary simultaneously. Depending upon the size of the design vector, the total number of architecture permutations can be very large. In such a case, it is practically impossible to enumerate, calculate, and compare all of the potential architectures. At this point, the systems engineer has two options. First, he may choose to – based on his intuition, experience, and understanding of





**Figure 3.11** TPF Cost Per Image vs. Orbit [Curtis et al, 1999]

the trade space of each subsystem – evaluate a small subset of the potential architectures. Different procedures, such as Taguchi's method which uses the principles of orthogonal arrays, exist to guide the systems engineer as to which architectural permutations should be tested to gain an understanding of the global trade space. The second option uses an automated optimization algorithm to systematically choose and evaluate alternative architectures with the intention of rapidly searching through the system trade space for global optima. Figure 3.12 illustrates how this might be done. Since the GINA model represents a system simulation with a set of inputs and outputs, it might be possible to loop an optimization algorithm around the simulation such that the knowledge gained from one set of outputs is used to select a new set of inputs (i.e. a new design vector) with the intention of optimizing the system architecture with respect to the metric(s) of interest to the customer. Such MDO algorithms are developed and applied in step four of the MMDOSA methodology, which is presented in Section 3.5.



**Figure 3.12** Wrapping an Optimization Algorithm Around a GINA Model

### 3.2.8 GINA Summary

Section 3.2 has provided a cursory introduction to the Generalized Information Network Analysis methodology for the comparative analysis of distributed satellite systems. In parallel with the explanation of each step in the process, the application of the GINA methodology to the conceptual design of a real DSS – NASA's Terrestrial Planet Finder mission – has been demonstrated. The power of GINA, which models and treats a space system as an information network, lies in its ability to allow the systems engineer to make meaningful, quantitative trades at the conceptual design level by directly relating lifecycle performance to lifecycle cost. The GINA model created here in step one of the MMDOSA methodology forms the foundation for the remaining steps in the methodology, beginning with step two. Notice how step two in the MMDOSA methodology – perform univariate trade studies – flows naturally from step six of the GINA methodology.

## 3.3 MMDOSA Step 2 – Perform Univariate Studies

Once a GINA model has been developed for the DSS and confidence in the output of the model has been established, the question arises of how to use the model to explore the DSS trade space. The first method of trade space exploration that systems engineers natu-

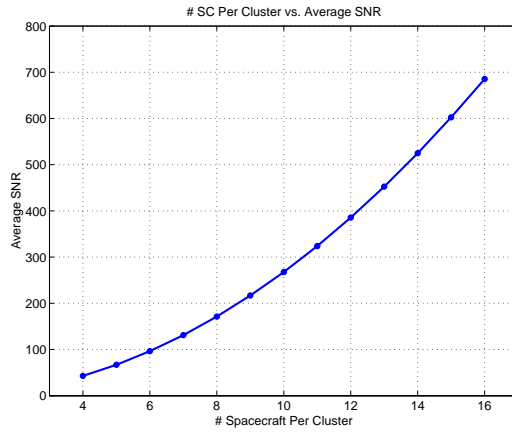
rally tend to use is to establish a baseline design, vary one parameter within the baseline design along a range of values, and see how the attributes of the design change [Shishko & Chamberlain, 1995]. This is known as a univariate study. Univariate studies investigate how a single system architecture parameter (i.e. simulation output variable or GINA metric) varies as a function of a single input parameter (i.e. design vector or constants vector variable). Univariate studies begin to give the systems engineer a feel for the trade space, and how different design decisions (i.e. different system architecture input parameters) affect the trade space.

Figure 3.13 illustrates three sample univariate studies on the TechSat 21 GINA model. The first graph (Figure 3.13a) plots the change in average SNR received from the target as a function of the number of spacecraft per cluster, all other parameters held constant. As expected from the radar physics equations presented in Chapter 5, SNR increases smoothly as the square of the number of TechSat 21 satellites within a single cluster. Thus, the systems engineer may safely conclude that SNR increases with cluster size. Figure 3.13b, which plots the probability of detection as a function of aperture diameter, everything else held constant, illustrates an example of a discontinuous relationship between a design variable and system attribute. Initially, increasing the radar transmission antenna aperture diameter has little effect on the system probability of detection. After passing a threshold value between 1 m and 1.5 m, however, the power-aperture product is suddenly large enough to provide an improvement in probability of detection that approaches the theoretical maximum of 100%. Thus, the systems engineer may conclude that threshold values exist for power-aperture product to achieve a reasonable probability of detection. Figure 3.13c plots the system lifecycle cost also as a function of aperture diameter, all other design parameters held constant. As expected from the phased array antenna CER, the lifecycle payload cost increases smoothly with aperture diameter. The lifecycle spacecraft bus cost also increases smoothly with aperture diameter (i.e. a larger satellite bus is required to support a larger payload), but at a smaller rate. Lifecycle launch costs, on the other hand, increase in a step-wise manner rather than continuously. This is because launch costs increase only when the total increase in satellite mass is large enough

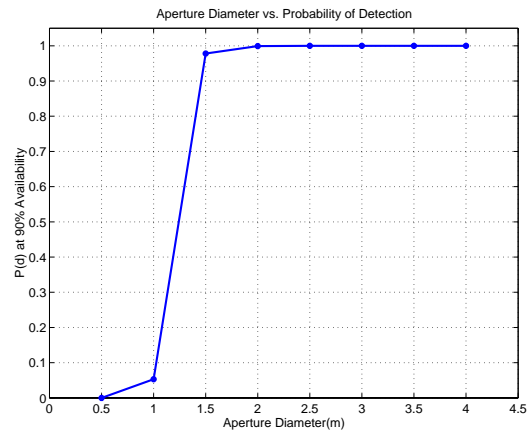
to change the solution of the launch vehicle selection optimization problem (Appendix B). Finally, operations costs are unaffected by aperture diameter. These three plots exemplify how univariate studies may be used to give the systems engineer a feel for the system trade space. Univariate studies may also be used to identify paradoxes and problems within the GINA model. For example, if the curve in Figure 3.13a displayed a negative rather than positive slope, then one would know that there was a mistake in the radar module because the results would not correlate with theory. Thus, univariate studies may also be used to verify and improve the fidelity of a GINA model. Chapters 4-6 contain the univariate studies carried out on the three DSS case studies.

Univariate studies may also be used to find the general shape of a "design family" within the system trade space. A design family is created by establishing a baseline design and then varying each parameter within the design vector across its entire range while holding all other components of the design vector at their baseline values. Table 3.9 lists the baseline design vector for the design family illustrated in Figure 3.14.

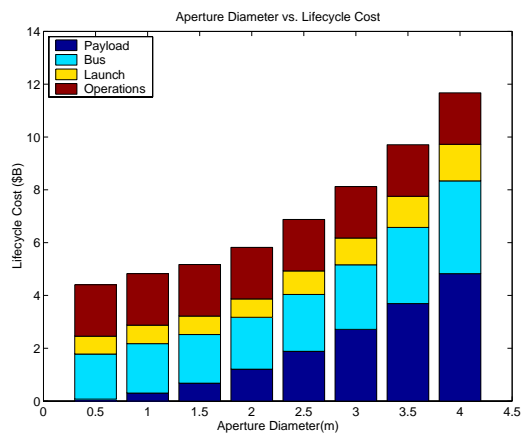
Figure 3.14 illustrates a sample TechSat 21 design family, with probability of detection (i.e. system performance) on the x-axis and lifecycle cost on the y-axis. Each dot represents a different TechSat 21 system architecture within the same design family. The black asterisk represents the baseline design (Table 3.9) for this particular family, and the diagonal lines are iso-CPF lines (i.e. lines of constant cost per unit probability of detection). Three separate types of relationships between the design vector parameters and system metrics may be identified within this plot. The first relationship is represented by the blue dots, which show how performance and lifecycle cost vary as the number of satellites per cluster increases from four to sixteen. Starting with the left-most blue dot, one sees that, initially, increasing the number of satellites per cluster provides a dramatic increase in lifecycle cost, but with only a small corresponding increase in system performance. Beyond seven satellites per cluster, lifecycle cost continues to increase at the same rate, but now there is also a dramatic improvement in system performance. Finally, beyond eleven satellites per cluster, lifecycle cost again continues to increase at the same rate, but now with



(a)



(b)



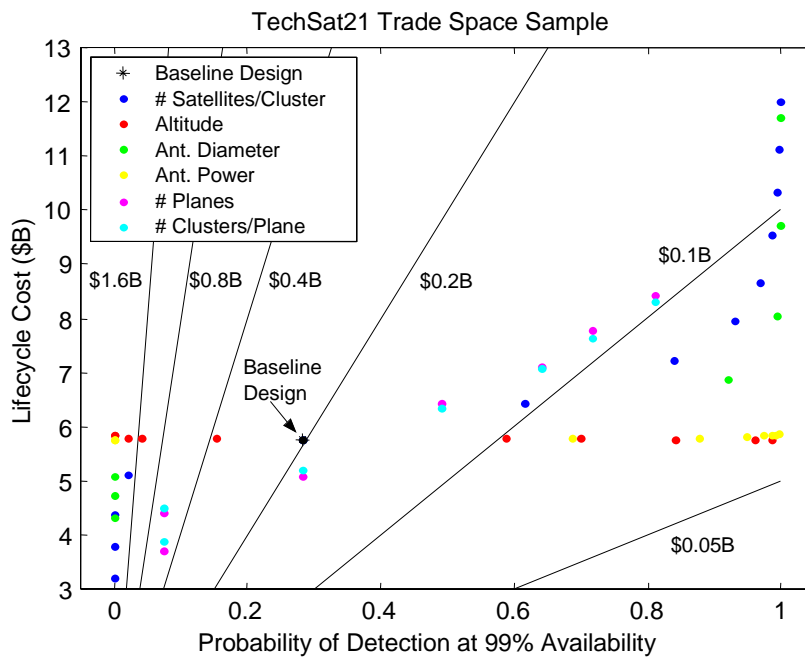
(c)

Figure 3.13 Sample TechSat 21 Univariate Studies

**TABLE 3.9** Baseline TechSat 21 Design Vector

Design Vector	Value
Satellites Per Cluster	8
Clusters Per Orbital Plane	6
Orbital Planes	6
Constellation Altitude	1000 km
Aperture Diameter	2 m
Radar Transmission Power	300 W

only a marginal improvement in system performance as it approaches its theoretical maximum of 100%. This results in the s-curve seen in the plot. Varying the aperture diameter (green dots) also produces a similar s-curve when plotted against system performance and lifecycle cost.



The second relationship is represented by the red dots, which show how performance and lifecycle cost vary as the altitude of the constellation increases from 500 km to 1500 km. Starting with the left-most red dot (1500 km), system performance improves as altitude

decreases, but apparently with no effect on lifecycle cost. In reality, lifecycle cost is decreasing with altitude as smaller, less costly launch vehicles are required to deploy and replenish the constellation. However, this cost savings is on the order of millions of dollars, and therefore does not show up well on the lifecycle cost scale in this plot, which is on the order of billions of dollars. A more pronounced negative slope would be seen if the range across which the altitude varied was greater than 1000 km. Varying the antenna transmission power (yellow dots) also produces a similar relationship in which system performance improves as transmission power increases, but with apparently no increase in lifecycle cost. In reality, lifecycle cost is increasing, but not with a magnitude observable in this plot due to the constant vector value of  $0.25 \text{ kg/m}^2$  for the areal density of the lithium ion solar arrays, which is the areal density the AFRL estimates will be available during the construction phase of this mission. Finally, both system performance and lifecycle cost improve linearly as the number of orbital planes (magenta dots) within the constellation and the number of clusters per orbital plane (cyan dots) increases.

To recap, univariate studies provide the systems engineer with a feel for the DSS trade space while also providing a means to identify potential problems with respect to model fidelity. While useful, univariate studies by themselves do not give the systems engineer a complete view of the DSS trade space. They fail to tell the entire story because they ignore the inherent design couplings that are present within every DSS conceptual design problem as discussed earlier in Chapter 1. In other words, because only one design variable is varied at a time, no insight is provided as to how the system attributes change when two or more design variables are varied simultaneously. In real-world design problems, systems engineers don't hold everything constant and vary only one parameter at a time, but rather trade multiple parameters against each other simultaneously. The real power of the GINA modeling approach lies in the ability of the system engineer to change multiple design parameters simultaneously (i.e. vary the entire design vector at once) and see how the system attributes change – how the global trade space varies. This will be done by the MDO algorithms developed in step 4 of the MMDOSA methodology. Before these algorithms can be used, however, one must first take a random sample of the trade space.

### 3.4 MMDOSA Step 3 – Take a Random Sample of the Trade Space

While univariate studies provide the systems engineer with a glimpse of a local trade space – the trade space local to the baseline architecture on which the univariate study is centered – such studies provide little insight into the global trade space of a DSS. To gain such insight, one must take a random sample of the global trade space. Such a random sample provides the systems engineer with preliminary data on how the system architecture metrics vary for different architectures throughout the global trade space and, in accordance with the principles of random sampling theory, provides critical data required to formulate the MDO DSS trade space search algorithms developed in Section 3.5.

#### 3.4.1 Random Sampling Theory

A common problem in statistics is how to assess a particular distribution without complete knowledge of the distribution. This leads to the field of statistical inference, which provides a methodology for deriving data on a *population* (i.e. the full set of architectures in the DSS trade space) after only measuring a small subset, or *sample*, of the population. A *random sample* is a sample collected in such a way that every unit in the population is equally likely to be selected [Bertsimas & Freund, 2000].

In many cases, it is impossible or impractical to measure the entire population. For distributed satellite systems, the computation time required to accurately assess a single design makes it impractical – due to both computational time and memory constraints – to simulate, measure, and evaluate the full DSS trade space. Therefore the systems engineer must depend on a random subset of measurements from the population (i.e. the complete trade space) to help make inferences concerning the global trade space. This requires an understanding of the notion of sampling.

Define the population as the complete global DSS trade space and a sample as a subset of the trade space. To avoid bias (i.e. consistently underestimating or overestimating some



characteristic of the population), a random sample of the population is required to ensure each measurement is statistically independent. The mathematical relationships presented in this section only apply if the population sample is truly random.

A sample is truly random when each member of the population has the same probability of being selected. Mathematically, if  $X_1, X_2, \dots, X_n$  are  $n$  independent random variables each having the same probability distribution  $f(x)$ , then  $X_1, X_2, \dots, X_n$  is a random sample of size  $n$  from the population  $f(x)$  and its joint probability distribution may be written as

$$f(x_1, x_2, \dots, x_n) = f(x_1)f(x_2) \cdots f(x_n) \quad (3.11)$$

Let  $\mu$  be the true population mean and  $\bar{X}$  the sample mean:

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n} \quad (3.12)$$

where  $n$  is the size of the random sample (i.e. the number of system architectures evaluated out of the complete DSS global trade space). When the random sample is arranged in increasing order of magnitude, the sample median  $\tilde{X}$  may be defined as:

$$\tilde{X} = \begin{cases} X_{\frac{(n+1)}{2}} & \text{if } n \text{ is odd} \\ \frac{X_{\frac{n}{2}} + X_{\frac{n}{2}+1}}{2} & \text{if } n \text{ is even} \end{cases} \quad (3.13)$$

The sample mode is the value in the sample that occurs with the greatest frequency. A sample mode may not necessarily exist and, if it does, it may not be unique. Finally, the range of a random sample is defined by the statistic  $X_l - X_s$  where  $X_l$  and  $X_s$  are the largest and smallest observations in the sample respectively.

Now focus on the sample mean and how accurately it represents the true population mean  $\mu$ . The sampling error  $w$  that occurs when  $\mu$  is estimated by  $\bar{X}$  is:

$$w = \bar{X} - \mu \quad (3.14)$$

This sampling error has a mean of zero. The variance of the sampling error depends upon the size  $n$  of the random sample and the spread (i.e. variance) of the entire population:

$$\sigma^2(w) = \frac{\sigma^2(x)}{n} \quad (3.15)$$

With knowledge of the mean and variance of the sampling error  $w$ , the Central Limit Theorem may now be used to help make inferences on the population mean [Barnett, 2001]. According to the Central Limit Theorem, when  $n$  is sufficiently large ( $n > 15$ ),  $w$  follows approximately a *normal* distribution. This outcome does *not* depend on the shape of the original population distribution. This means that if  $\bar{X}$  is the mean of a random sample of size  $n$  taken from a population with mean  $\mu$  and finite variance  $\sigma^2$ , then the limiting form of the distribution of

$$Z = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}} \quad (3.16)$$

as  $n \rightarrow \infty$  is the standard normal distribution  $n(z; 0, 1)$ . The three basic tenets of the Central Limit Theorem are [Barnett, 2001]:

1. The mean of the sampling distribution  $\bar{X}$  is identical to the population mean  $\mu$ .
2. The standard deviation  $v(x)$  of the distribution of sample means is equal to  $\sigma^2/n$ .
3. If  $n$  is large enough, the shape of the sampling distribution is approximately normal.

The systems engineer may now use the normal table to answer any probabilistic questions about the sampling error.

In reality, however, the systems engineer will not know the true population variance and therefore will not be able to compute the sampling error variance in Eqn. 3.15. This problem may be overcome by first computing the variance  $v^2$  of the random sample:

$$v^2(x) = \sum_{i=1}^n \frac{(x_i - \bar{X})^2}{(n-1)} \quad (3.17)$$

The sampling error variance may now be estimated as:

$$v^2(w) = \frac{v^2(x)}{n} \quad (3.18)$$

The probability distribution of the sampling error is:

$$w \approx N(0, v(w)) \quad (3.19)$$

With the estimate of the sampling error variance in Eqn. 3.18, the probability distribution of the population mean may be stated according to the Central Limit Theorem as:

$$\mu = N(\bar{X}, v(w)) \quad (3.20)$$

where  $v(w)$  is the sampling error standard deviation (i.e. the square root of the variance).

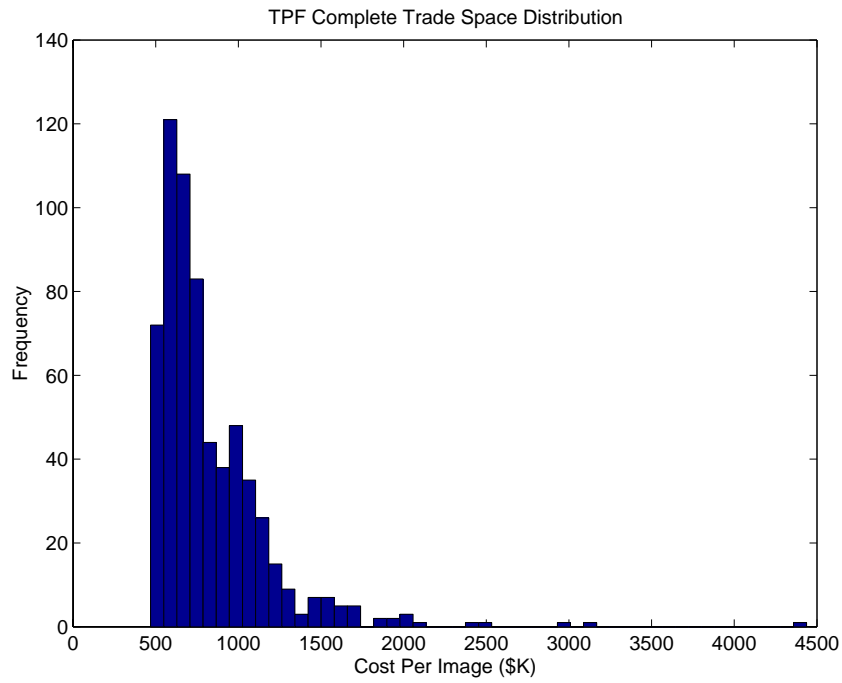
Perhaps more useful, a numerical range may now be defined. The 95% confidence interval for  $\mu$  based on a random sample of size  $n$  is [Barnett, 2001]:

$$95\% \text{ CI} = \bar{X} \pm \frac{2v(x)}{\sqrt{n}} \quad (3.21)$$

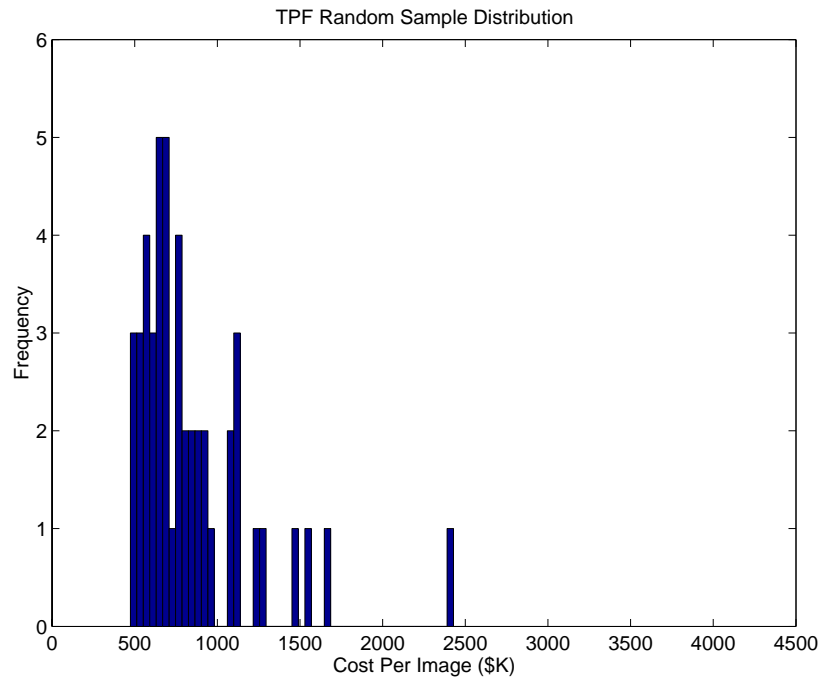
$$n \geq 15$$

### 3.4.2 TPF Example

To test the applicability of the above theory to the design of distributed satellite systems, a random sample of size  $n=48$  was taken from the TPF complete global trade space of 640 system architectures. Figure 3.15 plots the cost per image (CPI) metric distribution for both the complete global trade space and the random sample. At first glance, the shapes of the two distributions appear to be similar, and therefore the random sample appears to be a good approximation of the complete trade space.



(a)



(b)

**Figure 3.15** TPF CPI Histograms for the Complete Global Trade Space (a) and the Random Sample of 48 Architectures (b)

Table 3.10 compares the statistics between the complete trade space and the random sample. In this case, the random sample represents 7.5% of the complete trade space. When a random sample is such a large percentage of the population, the results provided by random sampling theory should be highly accurate. This appears to be the case as the sample mean differs from  $\mu$  by only \$22K/image. While there exists a large discrepancy between the sample maximum CPI and the population maximum CPI, the sample median and minimum more closely approximate those of the complete global trade space.

**TABLE 3.10** TPF Population and Random Sample CPI Statistics

Parameter	Complete Global Trade Space	Random Sample
Size ( $n$ )	640	48
Mean (\$K/image)	827	849
Standard Deviation (\$K/image)	357	367
Maximum (\$K/image)	4438	2429
Median (\$K/image)	723	734
Minimum (\$K/image)	469.6	473.9

Using the equations from the previous section, the sampling error variance  $v^2(w)$  is 2806 (\$K/image)<sup>2</sup> and the sampling error standard deviation  $v(w)$  is 53.0 \$K/image. Thus, the sampling error may be represented as a zero mean normal distribution:  $w=N(0, 53.0)$ . The 95% confidence interval for  $\mu$  based on the random sample of the TPF trade space is:

$$743\$K/image \leq \mu \leq 955\$K/image \quad (3.22)$$

The true value of  $\mu$  (827\$K/image) falls within this interval. Thus, the random sample does accurately predict the population mean of the DSS metric(s) of choice, in this case the TPF CPI, after evaluating only a subset of the complete trade space. Used this way, the random sample tells the systems engineer how an "average" DSS architecture for a given mission will rate with respect to the metric(s) of interest. While interesting, this aspect of random sampling does not really help the systems engineer to optimize the sys-

tem architecture. The next section discusses how the information learned from the random sample aids in the optimization of the conceptual design.

The minimum and maximum for each metric measured in the sample provide initial bounds which should be surpassed after an MDO algorithm has been applied to search the trade space. For example, Table 3.10 lists that the minimum CPI architecture in the random sample is 483.0 \$K/image and the minimum CPI architecture in the global trade space is 469.6 \$K/image. Therefore 483.0 \$K/image provides an initial lower bound on the CPI metric that the systems engineer expects the MDO algorithm to improve upon.

Furthermore, the data gathered from the random sample may be used to tailor the structure of the MDO algorithms developed in Section 3.5. The importance of the  $\Delta$  parameter within the simulated annealing algorithm is discussed in Section 3.5. Ideally, this parameter would be computed with knowledge of the complete distribution of each metric which the DSS is being optimized with respect to. When exploring a large trade space, however, the  $\Delta$  parameter must be computed from the information gained by taking a random sample of the global trade space. Please see Section 3.5.2 for details on how these statistics from the random sample are used to tailor the MDO search algorithms.

### 3.4.3 Summary

To summarize, random sampling provides the systems engineer with an glimpse of the global trade space, with initial bounds on each of the metrics the design will be optimized with respect to, and statistical data that will be used to tailor the MDO algorithms. This section has covered the mathematics and mechanics of random sampling, illustrated the use of random sampling on two of the case studies, and discussed how random sampling statistics will be used by the remainder of the MMDOSA methodology. The multiobjective, multidisciplinary design optimization algorithms are now ready to be developed.

## **3.5 MMDOSA Step 4 – Formulate and Apply MDO Algorithm(s)**

Multiobjective, multidisciplinary design optimization algorithms form the core of the MMDOSA methodology. These algorithms help the systems engineer find the best architectures for a DSS based on the metric(s) of interest to the customer. The trade space for a DSS can be enormous – too large to enumerate, analyze, and compare all possible architectures. Furthermore, a design that is globally optimized on the basis of a system metric(s) is likely to vary drastically from a design that has been developed by optimizing each subsystem. If used properly, MDO algorithms should enable the systems engineer to globally optimize a DSS without having to resort to complete enumeration of the trade space. This section develops and analyzes the performance of several different MDO algorithms on both single objective and multiobjective DSS conceptual design problems.

### **3.5.1 Single Objective Optimization**

In a single objective optimization problem, one wishes to find the best DSS architecture on the basis of a single metric. The results from a single objective optimization exercise should yield the most promising family(s) of system architectures to be carried forward into the next phase of the design process. This section develops, analyzes, and compares the performance of several different single objective optimization techniques on the single objective DSS conceptual design problem. At the conclusion of this section, the best MDO algorithms for the single objective DSS conceptual design optimization problem will have been identified.

#### **Comparison of MDO Techniques**

Four separate single objective multidisciplinary design optimization techniques were investigated for their applicability to the conceptual design of distributed satellite systems. The four multidisciplinary design optimization techniques tested were Taguchi's method, heuristic simulated annealing, a pseudogradient search, and a univariate search. Each MDO technique was applied to the problem of developing a system architecture that min-

imizes the cost per image metric for the TPF mission. The best conceptual design solution obtained from each technique after evaluating only 7.3%-7.8% of all possible design solutions was compared to the true optimal solution obtained from complete enumeration. It was found that while the simulated annealing, pseudogradient search, and univariate search algorithms all found the true optimal solution at least once over multiple trials, simulated annealing did so with the greatest degree of confidence statistically.

The TPF mission is currently in the conceptual design phase, and several widely varying conceptual design architectures ranging from structurally connected to tethered to separated spacecraft arrays have been proposed. For the purpose of this case study, four design parameters are isolated as the key independent variables in the design problem - heliocentric orbital radius ( $\gamma_1$ ), collector connectivity/geometry ( $\gamma_2$ ), number of collector apertures ( $\gamma_3$ ), and collector aperture diameter ( $\gamma_4$ ). Together, these four parameters make up the TPF design vector  $\Gamma$ :

$$\Gamma = [\gamma_1 \ \gamma_2 \ \gamma_3 \ \gamma_4] \quad (3.23)$$

Table 3.11 lists the range of discrete values considered for each design vector variable  $\gamma$  in this case study. Each design vector defines a unique TPF mission architecture. The design vector was kept small to allow for the use of complete enumeration to verify the global optimum. Taking into account every possible combination of the variables in Table 3.11, the trade space of this study contains 640 different design vectors, or 640 unique TPF system architectures. This set of architectures is defined as the full-factorial trade space.

A cost per function metric will be used to evaluate the strength of each proposed TPF design architecture. For this mission, the CPF is defined as the cost per image (CPI). The objective for each of the MDO techniques is to find the TPF architecture that minimizes the CPI. The only constraints on the problem are those on the capability isolation and integrity requirements and the allowable values for each variable in the design vector. The TPF conceptual design optimization problem may now be represented mathematically as:



**TABLE 3.11** The TPF Design Vector

Design Variable	Design Vector Parameter	Range	Discretization Units
Heliocentric Orbital Radius	$\gamma_1$	1.0 AU - 5.5 AU	0.5 AU
Collector Connectivity/Geometry	$\gamma_2$	SCI-1D, SCI-2D, SSI-1D, SSI-2D	Unique
Number of Collector Apertures	$\gamma_3$	4 - 10	2
Diameter of Collector Apertures	$\gamma_4$	1 m - 4 m	1 m
<b>Total # Permutations</b>		<b>640</b>	

$$\begin{aligned}
 \text{Objective:} & \quad \text{Min} \left( \frac{\sum_{y=1}^5 \Phi_y(\Gamma)}{\sum_{y=1}^5 \Psi_y(\Gamma)} \right) \\
 \text{Constraints:} & \quad \text{Subject to} \\
 \text{Isolation} & \quad \theta_r \geq 20 \text{ milliarc sec} \\
 & \quad \Omega \leq 10^{-6} \\
 \text{Integrity} & \quad \text{SNR}_s \geq 5 \\
 & \quad \text{SNR}_{ms} \geq 10 \\
 & \quad \text{SNR}_{ds} \geq 25 \\
 \text{Availability} & \quad A_v \geq 0.95 \\
 \text{Bounds} & \quad 1.0\text{AU} \leq \gamma_1 \leq 5.5 \text{ AU} \\
 & \quad 4 \leq \gamma_3 \leq 10 \\
 & \quad 1 \text{ m} \leq \gamma_4 \leq 4 \text{ m}
 \end{aligned} \tag{3.24}$$

where  $y$  is the year of the mission,  $\Phi$  is cost,  $\Psi$  is the number of images (i.e. surveys + medium spectroscopies + deep spectroscopies),  $\theta_r$  is angular resolution,  $\Omega$  is null depth,  $\text{SNR}_s$  is the survey mode signal-to-noise ratio (i.e. photon count received from the planet over the photon count received from all noise sources),  $\text{SNR}_{ms}$  is the medium spectroscopy mode signal-to-noise ratio,  $\text{SNR}_{ds}$  is the deep spectroscopy mode signal-to-noise

ratio, and  $A_v$  is the system availability. For a complete description of the GINA model behind this formulation of the TPF conceptual design problem, please see Chapter 4.

DSS designs are discrete, nonlinear combinatorial problems – each design variable can only take on a set number of unique values. For this reason, continuous linear techniques like the simplex method cannot be used. For this MDO comparison study, the chosen optimization methods are Taguchi's method, simulated annealing, pseudo-gradient search, and univariate search. The following sections provide an overview of each technique, including the origin of the technique, a summary explanation of how the algorithm works, and the specifics on how the optimization technique was applied to the TPF conceptual design problem.

### **Taguchi Methods**

Beginning in the late 1970s, Professor Genichi Taguchi of Japan created a framework that uses experimental design to improve the quality control of manufacturing processes [Logothetis and Wynn, 1989]. Orthogonal arrays are used to define a set of experiments with different combinations of design variable settings. In an orthogonal test matrix, each value of each design variable is tested an equal number of times, and each of these values is tested with every value of all other design variables an equal number of times [ASQC, 1983]. While Taguchi methods have been used to improve the quality of manufacturing designs for many years now, these methods have only recently been considered for application to the design of aerospace systems. For example, Taguchi analysis has been used by engineers at the NASA Langley Research Center to create a Pareto optimal design curve for a tetrahedral space truss platform by evaluating only 31 of the possible 19,683 truss designs [Wu and Lake, 1996].

The size of the orthogonal design experiment matrix is a function of the total number of design variables and the number of different values each design variable may assume. Each row in this matrix represents a single experiment with a unique combination of design variable settings (i.e. a unique design vector input to the GINA model). The result

of each experiment is converted into a parameter Taguchi defines as the signal-to-noise ratio (SNR), which combines the loss effects attributable to both missing a target and having too large a variation about that target [Logothetis and Wynn, 1989]. It should be noted that this SNR is different from the SNR represented in Eqn. 3.24. In the case where the goal is to minimize the output of the metric of interest in each experiment, the SNR is defined as:

$$\text{SNR} = -10 \log_{10} \left( \frac{1}{N} \sum z^2 \right) \quad (3.25)$$

where  $N$  is the total number of samples taken during an experiment and  $z$  is the metric output of each sample of that particular experiment [Ross, 1996].

After the experiments for every row in the orthogonal matrix have been completed and converted to SNR's, the balanced mean  $m$  for all of the experiments  $n$  is calculated:

$$m = \frac{1}{n} \sum_{i=1}^n \eta_i \quad (3.26)$$

where  $\eta_i$  is the SNR of each experiment  $i$ . Next, the main effect  $m_{\gamma x}$  of each design vector variable  $\gamma$  at a particular setting  $x$  is computed:

$$m_{\gamma x} = \frac{1}{n_{\gamma x}} \sum_{i=1}^{n_{\gamma x}} \eta_i \quad (3.27)$$

The most probable optimal design is then found by choosing the setting for each design variable that leads to the largest SNR. Finally, the performance  $\eta_{opt}$  of the proposed optimal design may be predicted:

$$\eta_{opt} = m + \sum_{\gamma_i=1}^d (m_{\gamma_{ix}} - m) \quad (3.28)$$

where  $d$  represents the total number of design variables in the experiment and  $x$  denotes the proposed optimal setting for each design variable.

For the TPF case study, a  $48 \times 4$  orthogonal test matrix was required to properly conduct the Taguchi analysis given the nature of the TPF design vector. Notice that this required the use of two heliocentric orbital radii – 6 AU and 6.5 AU – that were not in the original feasible space of the design vector. In the Taguchi analysis, however, their presence is required to ensure the statistical independence of the three other design vector elements. Because the "experiments" are computer simulations (i.e. runs of the GINA model with different design vectors but identical constants vectors), the value of  $N$  in Eqn. 3.25 is one, and therefore the actual measurement rather than the SNR equivalent can be used in Eqn. 3.26 through Eqn. 3.28. Table 3.12 illustrates the orthogonal test matrix for the TPF case study.

### Simulated Annealing

Simulated annealing is a heuristic technique developed in the early 1980's that mathematically mirrors the cooling of a material to a state of minimum energy [Kirkpatrick et al, 1983]. If a material cools too quickly, the crystals within the material will harden in a sub-optimal configuration. Likewise, the premise behind the simulated annealing algorithm is the assertion that if a solution is converged upon too quickly, that solution will be suboptimal. Simulated annealing was chosen over the other two well known heuristics of genetic algorithms and the tabu search due to its relative ease of implementation, lower computational requirements, and better success rate in finding the optimal solution to similar complex problems [Brooks et al, 1998].

First, the objective function, or system energy,  $E(\Gamma)$  to be minimized by the algorithm must be defined, where  $\Gamma$  is the design vector:

$$E(\Gamma) = f(\gamma_1 \ \gamma_2 \ \cdots \ \gamma_n) \quad (3.29)$$

**TABLE 3.12** Orthogonal Taguchi Analysis Test Matrix for the TPF Case Study

Case	Orbit (AU)	Collector Connectivity/ Geometry	Number of Apertures	Aperture Diameter (m)
1	1.0	SCI-1D	4	1
2	1.0	SCI-2D	6	2
3	1.0	SSI-1D	8	3
4	1.0	SSI-2D	10	4
5	1.5	SCI-1D	6	3
6	1.5	SCI-2D	4	4
7	1.5	SSI-1D	10	1
8	1.5	SSI-2D	8	2
9	2.0	SCI-1D	8	4
10	2.0	SCI-2D	10	3
11	2.0	SSI-1D	4	2
12	2.0	SSI-2D	6	1
13	2.5	SCI-1D	10	2
14	2.5	SCI-2D	8	1
15	2.5	SSI-1D	6	4
16	2.5	SSI-2D	4	3
17	3.0	SCI-1D	4	1
18	3.0	SCI-2D	6	2
19	3.0	SSI-1D	8	3
20	3.0	SSI-2D	10	4
21	3.5	SCI-1D	6	3
22	3.5	SCI-2D	4	4
23	3.5	SSI-1D	10	1
24	3.5	SSI-2D	8	2
25	4.0	SCI-1D	8	4
26	4.0	SCI-2D	10	3
27	4.0	SSI-1D	4	2
28	4.0	SSI-2D	6	1
29	4.5	SCI-1D	10	2
30	4.5	SCI-2D	8	1
31	4.5	SSI-1D	6	4
32	4.5	SSI-2D	4	3
33	5.0	SCI-1D	4	1
34	5.0	SCI-2D	6	2
35	5.0	SSI-1D	8	3
36	5.0	SSI-2D	10	4
37	5.5	SCI-1D	6	3
38	5.5	SCI-2D	4	4
39	5.5	SSI-1D	10	1
40	5.5	SSI-2D	8	2
41	6.0	SCI-1D	8	4
42	6.0	SCI-2D	10	3
43	6.0	SSI-1D	4	2
44	6.0	SSI-2D	6	1
45	6.5	SCI-1D	10	2
46	6.5	SCI-2D	8	1
47	6.5	SSI-1D	6	4
48	6.5	SSI-2D	4	3

Each  $\gamma$  represents a different DSS design vector variable, such as the orbital altitude or total number of satellites. The algorithm begins with an initial state vector  $\Gamma_i$  containing randomly chosen values for each design vector variable within the bounds placed on that variable. The value of the objective function  $E(\Gamma_i)$  is then computed for this design vector. Next, the design vector is randomly perturbed to find a new design vector ( $\Gamma_{i+1}$ ) in the "neighborhood" – where a neighbor is a design vector which shares some identical entries with the original design vector – of the current design vector, and the new  $E(\Gamma_{i+1})$  is computed. If  $E(\Gamma_{i+1}) < E(\Gamma_i)$ , then  $\Gamma_{i+1}$  is accepted as the new design vector. If  $E(\Gamma_{i+1}) > E(\Gamma_i)$ ,  $\Gamma_{i+1}$  can still be accepted as the new design vector with a probability:

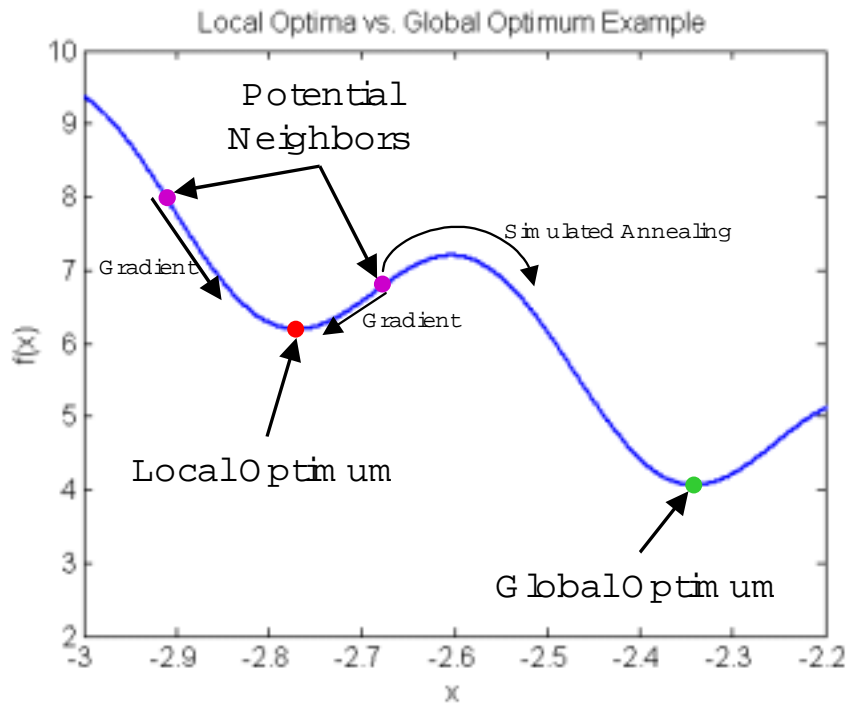
$$\text{Prob}(E) = e^{-\frac{\Delta}{T}} \quad (3.30)$$

where

$$\Delta = E(\Gamma_{i+1}) - E(\Gamma_i) \quad (3.31)$$

$T$  is the system temperature, a parameter that decreases as the number of iterations in the optimization increases. Thus, the likelihood of accepting a design vector with a greater system energy decreases over time. The ability to accept a less optimal  $\Gamma_{i+1}$  reduces the chance that the algorithm's solution will become trapped in local minima (Figure 3.16). This procedure is repeated until no new solutions are accepted after a specified number of iterations or until the algorithm is manually terminated. In theory, the algorithm converges to the optimal solution as the system temperature lowers towards zero. Unlike linear or integer programming methods, however, heuristic algorithm solutions are not guaranteed to be globally optimal.

Table 3.13 summarizes the steps in the simulated annealing algorithm. For the TPF case study, cost per image replaces system energy as the metric to be minimized. Initial system temperature  $T_i$  was set to 1000, and was multiplied by a factor of 0.85 after each iteration. The algorithm was terminated after the completion of 48 iterations.



**Figure 3.16** How the Simulated Annealing Algorithm Escapes Local Optima

**TABLE 3.13** The Simulated Annealing Algorithm

Step	Description
1	Choose a random design vector $\Gamma_i$ , select initial system temperature, and outline cooling schedule.
2	Evaluate $E(\Gamma_i)$ via the GINA simulation model.
3	Perturb the current design vector $\Gamma_i$ to obtain a neighboring design vector $\Gamma_{i+1}$ .
4	Evaluate $E(\Gamma_{i+1})$ via the GINA simulation model.
5	If $E(\Gamma_{i+1}) < E(\Gamma_i)$ , $\Gamma_{i+1}$ is the new current design vector.
6	If $E(\Gamma_{i+1}) > E(\Gamma_i)$ , then accept $\Gamma_{i+1}$ as the new current design vector with a probability $e^{-\Delta/T}$ where $\Delta = E(\Gamma_{i+1}) - E(\Gamma_i)$ . Otherwise, $\Gamma_i$ remains the current design vector.
7	Reduce system temperature according to the cooling schedule.
8	Repeat steps 3 - 7 until the algorithm terminates.

### Pseudogradient Search

The French mathematician Pierre Fermat introduced some of the first formal ideas in the field of optimization, including the observation that at the maximum and minimum of a function the tangent line of any curve would be horizontal. Fermat’s observation was

implemented mathematically to find maxima and minima through the derivative operation. Even functions with multiple variables could now be optimized through the use of partial derivatives.

Unfortunately, DSS conceptual design problems are discrete rather than continuous. This lack of continuity violates calculus' Extreme Value Theorem, and thus prevents the use of derivatives to find designs that maximize or minimize the metric of choice.

Finite differencing, however, may be used to approximate the sensitivity of a particular point design to a change in each element of the design vector:

$$\delta(m) = f(\Gamma_{i+1}) - f(\Gamma_i) \quad (3.32)$$

where  $\delta$  is the difference in the metric  $m$  of choice resulting from a unit change in one element  $\gamma_i$  of the design vector  $\Gamma_i$ . The gradient of the metric  $\nabla(m)$  with respect to that element of the design vector is then:

$$\nabla(m) = \frac{\delta}{\gamma_{i+1} - \gamma_i} \quad (3.33)$$

where  $\gamma$  is the element of the design vector that was perturbed. Depending upon the slope of the gradient, a new design vector may be defined by moving the variable about which the gradient was computed in the direction that either increases or decreases the design metric, depending upon whether the goal of the single objective problem is maximization or minimization. This modified version of the gradient search works on the assumption that the output metric is approximated to be piecewise linear when only a single element of the design vector changes. Because the DSS conceptual design problem is neither convex nor concave in general, however, this algorithm may get trapped in local extrema.

For the TPF case study, an initial design vector was chosen by a random number generator. The gradient of the CPI metric was then computed about the heliocentric orbital radius, number of apertures, and aperture diameter elements of the design vector by changing each of these elements by one unit and then performing the finite differencing



calculation. For the collector connectivity/geometry ( $\gamma_2$ ) element of the design vector, the gradient of the CPI metric was computed by performing a finite difference for all three remaining architectures. A new design vector is then selected by perturbing each element of the design vector one unit in the direction with the largest negative CPI gradient. Thus, one run of the gradient search requires the evaluation of seven separate TPF architectures – six for finite differencing and one to evaluate the new design vector. For the TPF case study, this algorithm was applied seven successive times starting from the randomly generated initial design vector, resulting in the evaluation of 50 design vectors before arriving at the best solution. Because the generation of the initial design vector is a stochastic process, the gradient search algorithm was applied multiple times with multiple different randomly generated initial design vectors.

### Univariate Search

The final single objective MDO method explored for DSS conceptual design – the univariate search – is a common practice for initially exploring the trade space of a complex problem, as introduced in Section 3.3. First, a baseline design vector is chosen. Next, a single element  $\gamma_i$  of the design vector  $\Gamma$  is varied over its entire allowable range, while all of the remaining design vector elements are held constant at their baseline values. If the goal is objective function minimization, the new value for  $\gamma_i$  in the new design vector  $\Gamma_{i+1}$  is the value that produced the minimum objective value over the examined range. This process is repeated for each element of the design vector, with the other non-varying  $\gamma_i$  maintaining their original baseline values until every element in the design vector has been evaluated over its allowable range. The new design vector  $\Gamma_{i+1}$  is then created by taking the value of each  $\gamma_i$  that minimized the objective function. If desired,  $\Gamma_{i+1}$  may be used as a new baseline design vector about which to repeat this process of exploring each axis of each design vector variable individually.

This algorithm has many potential drawbacks. First, it is computationally expensive as every value of each element of the design vector must be tested before a new solution is converged upon. Second, this approach fails to capture the coupled effects that combina-

tions of design vector elements can have on the objective function. Finally, the solution depends upon the initial baseline design vector, and the final solution is not necessarily the global optimum. Nevertheless, the approach should work fairly well for distributed satellite systems with shallow trade spaces. Additionally, the univariate search algorithm provides the benefit of giving the systems engineer a feel for the trade space - such as how the performance of a system varies as the orbit varies, all other things held constant.

For the TPF case study, the baseline design vector was chosen by a random number generator. The univariate search algorithm was then executed over two successive runs, with each run requiring the evaluation of 22 separate design vectors, to converge upon a final solution. Since the generation of the initial baseline design vector is a stochastic process, this algorithm was applied multiple times from multiple different randomly generated initial baseline design vectors.

### **Results – Comparison of Single Objective MDO Techniques**

After each of the four MDO techniques have been applied to the TPF conceptual design problem, their performance at finding good solutions after evaluating less than 8% of the total system trade space may be analyzed. Both the Taguchi analysis and the simulated annealing algorithm evaluated 48 TPF design vectors to arrive at a solution. The pseudogradient search and univariate search techniques evaluated 50 and 47 design vectors respectively before converging upon a solution. Unfortunately, it was not possible to have all four MDO methods evaluate an identical number of design vectors before solution convergence because the pseudogradient search and single axis exploration techniques converge on a new solution after the evaluation of every seven and twenty-two design vectors respectively. However, a solid comparison between the effectiveness of all four MDO methods may still be made as the number of design vector evaluations for solution convergence are within 0.5% of each other – the univariate search technique tests 7.3% of the 640 possible TPF architectures at the low extreme while the pseudogradient technique searches 7.8% of the TPF architectures at the high extreme.

To determine the accuracy of these four MDO methods, the CPI metric was computed for all 640 full-factorial TPF design vectors. By comparing the best solution from each MDO method with the true optimal solution found by complete enumeration, the performance of each MDO technique can be quantitatively assessed. While complete enumeration to find the true optimal solution was possible for this controlled experiment, an expansion of the design vector  $\Gamma$ , both by including more independent variables  $\gamma_i$  and considering more possible values for each independent variable, could quickly make full enumeration of all potential design vectors a computationally impractical approach. Rather, the results of this work should identify which MDO techniques can quickly find good solutions by evaluating a minimum number of alternative design vectors for large, complex DSS conceptual design problems.

From complete enumeration of all 640 possible design vectors, the true optimal solution to the design problem was found to be an architecture with a structurally connected two-dimensional configuration located at 4 AU, with eight collector apertures, each of which is 4 m in diameter. This TPF architecture minimized the CPI metric at a value of \$469.6 thousand dollars per image.

The mean CPI for the 48 TPF design vectors evaluated in the Taguchi analysis matrix of Table 3.12 was \$818.2 thousand. After converting each CPI to an SNR, the balanced mean was computed from Eqn. 3.26 to be -57.8 dB. The optimized TPF design architecture is determined by setting each of the four design vector variables to the value that yields the largest SNR shown in boldface in Table 3.12. In this manner, Taguchi's method converges upon a structurally connected two-dimensional interferometer located at 3 AU with six collector apertures, each 4 m in diameter, as the best design.

Table 3.14 and Table 3.15 illustrate the results from the Taguchi analysis. Using Eqn. 3.28, the predicted CPI of this architecture is \$465.1 thousand. The actual performance of this architecture was evaluated by entering the Taguchi predicted best TPF design vector into the GINA model, which computed the true CPI to be \$499.1 thousand. Thus, the CPI

**TABLE 3.14** Taguchi Analysis Results

Orbit (AU)	Collector Connectivity/ Geometry		Number of Collector Apertures		Collector Diameter (m)		
	SNR	SNR	SNR	SNR	SNR	SNR	
1.0	-60.7	SCI		4	-59.9	1	-60.8
1.5	-57.4	1D	-57.6	<b>6</b>	<b>-57.0</b>	2	-58.0
2.0	-57.8	<b>2D</b>	<b>-57.1</b>	8	-57.2	3	-56.7
2.5	-57.9			10	-57.1	<b>4</b>	<b>-55.7</b>
<b>3.0</b>	<b>-57.0</b>	SSI					
3.5	-57.1	1D	-57.9				
4.0	-57.2	2D	-58.6				
4.5	-58.3						
5.0	-57.4						
5.5	-57.3						
6.0	-57.4						
6.5	-58.5						

of the Taguchi predicted best architecture differs from the true optimal architecture by 6.1%. This discrepancy illustrates how the nonlinear nature of the DSS conceptual design problem may hinder the effectiveness of Taguchi analysis techniques. Inspection of the design vectors reveals that Taguchi's method did converge upon the correct optimal values for the collector connectivity/geometry and aperture diameter, but failed to select the optimal orbit and number of collector apertures.

**TABLE 3.15** Optimized Configuration From Taguchi Analysis

Parameter	Taguchi Value	Optimal Value
Orbit (AU)	3	4
Collector Connectivity/Geometry	SCI-2D	SCI-2D
Number of Collector Apertures	6	8
Collector Diameter (m)	4	4
Taguchi Predicted Optimum Architecture CPI (\$K)	465.1	
Actual CPI (\$K) of Taguchi Predicted Optimum	499.1	
Taguchi Error from Predicted Optimum	7.1%	
True Optimum CPI (\$K)	469.6	
Taguchi Error from True Optimum	6.1%	

Because the three remaining MDO techniques may vary as a function of their starting point, are probabilistic in nature, and can converge upon local minima; multiple trials were executed for each technique. Each trial began with a different, randomly generated initial design vector. Table 3.16 illustrates the results of ten trials for each algorithm. Both the pseudogradient search and univariate search algorithms converged upon the true optimal solution four out of ten times, while the simulated annealing algorithm converged on the true optimal solution only once out of ten times. However, the heuristic simulated annealing approach exhibited the lowest mean square error (*MSE*) between the algorithm solutions and the true optimal solution, and thus exhibits better performance than the pseudogradient and univariate search algorithms in terms of consistently finding the best TPF system architectures. The mean square error, which measures how well each algorithm consistently finds the best solutions, may be computed as:

$$MSE = \frac{\sum_{i=1}^n (CPI_i - CPI^*)^2}{n} \quad (3.34)$$

where each value of  $i$  represents a single trial of the MDO algorithm,  $n$  is the total number of trials,  $CPI_i$  is the cost per image of the best system architecture from trial  $i$ , and  $CPI^*$  is the cost per image of the optimal TPF architecture within the global system trade space.

### **Analysis of MDO Techniques Comparison Results Within the Context of the TPF Trade Space**

Three of the four investigated MDO techniques – simulated annealing, pseudogradient search, and univariate search – found the true optimal solution for the TPF conceptual design problem. The remaining technique, Taguchi analysis, converged upon a solution with a CPI within 6.1% of the true optimal solution. The plot in Figure 3.17a illustrates the full-factorial trade space for the TPF conceptual design case study. The dashed diagonal lines represent lines of iso-CPI (i.e. lines of constant cost per image). All the system architectures that fall on these lines exhibit the same cost-effectiveness and thus are identi-

**TABLE 3.16** Simulated Annealing, Pseudogradient Search, and Univariate Search Algorithm Results

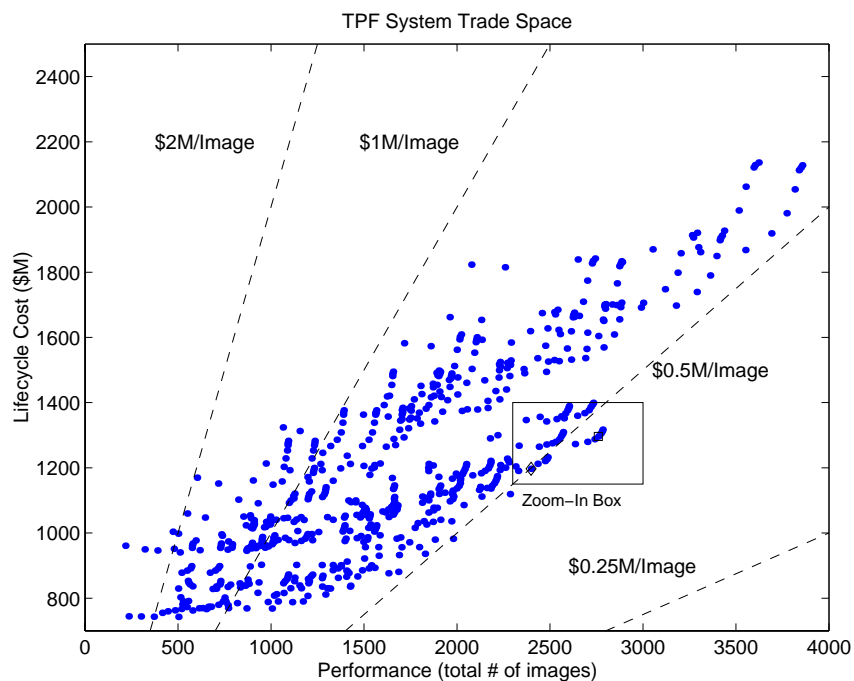
Trial	Simulated Annealing CPI (\$K)	Pseudogradient Search CPI (\$K)	Univariate Search CPI (\$K)
1	493.8	470.8	<b>469.6</b>
2	470.0	470.0	<b>469.6</b>
3	<b>469.6</b>	<b>469.6</b>	513.8
4	505.1	520.9	<b>469.6</b>
5	470.8	<b>469.6</b>	526.0
6	470.8	<b>469.6</b>	513.8
7	493.8	532.6	470.8
8	470.0	<b>469.6</b>	525.9
9	498.2	470.8	<b>469.6</b>
10	496.4	483.0	473.9
<b>MSE*</b> <b>(\$K/Image)<sup>2</sup></b>	<b>397.1</b>	<b>678.3</b>	<b>1027.8</b>

\*True Optimal Solution=\$**469.6K**

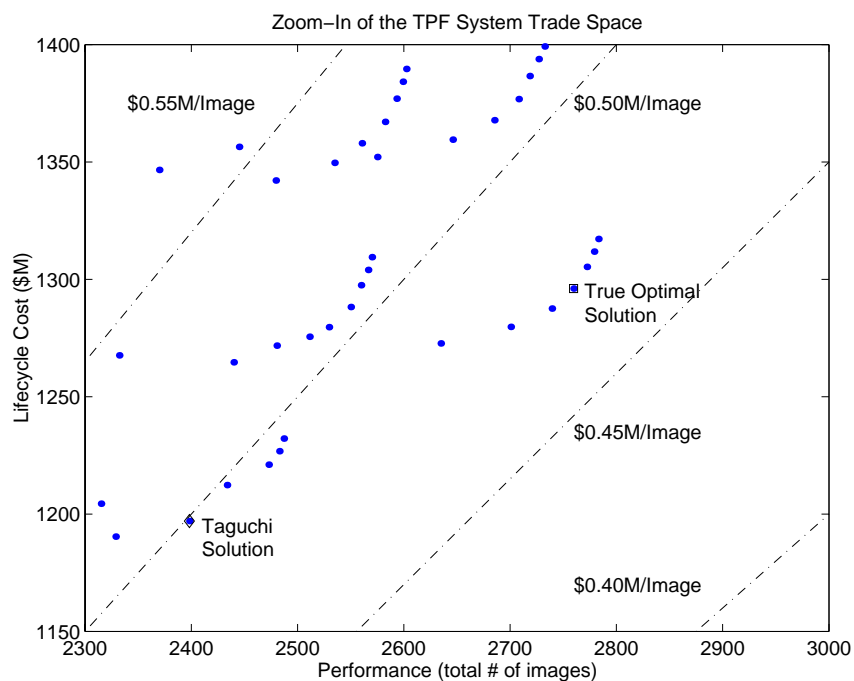
cal from the perspective of the objective function in Eqn. 3.24, albeit with different levels of performance and lifecycle cost.

The zoom-in plot in Figure 3.17b shows the location of the true and Taguchi optimal solutions in the trade space. According to the definitions of good solutions and best solutions outlined in Table 2.5, Taguchi's method found a good solution and the remaining three methods found best solutions at their peak performance on the basis of the CPI metric after evaluating less than 8% of the total trade space. While complete enumeration guarantees optimality and was possible in this case, the geometric growth rate in the number of system architectures as a function of the design vector size makes an exhaustive search impractical for large problems. These results illustrate how MDO methods may be successfully applied to large single objective optimization problems to efficiently find the best system architectures during the conceptual design of a DSS.

Also notice in the zoom-in plot in Figure 3.17b the existence of small arcs within the trade space. Each arc may be thought of as a separate family of system architectures within the trade space. Also notice how each arc contains a local minima with respect to the CPI metric. These local minima affect the solution of the DSS conceptual design optimization



(a)



(b)

**Figure 3.17** TPF Full-Factorial Trade Space (a) and Trade Space Zoom-In (b) with the True and Taguchi Optimal Solutions

problem as discussed in Section 2.2.2. In this case, the poor performance of the pseudo-gradient and univariate search algorithms as measured in Table 3.16 appears to be due to the susceptibility of these algorithms getting stuck in these local minima. Conversely, heuristics techniques like simulated annealing are intended for nonconvex problems with multiple local minima, and thus the simulated annealing algorithm performs the best as measured by the mean square error metric in Table 3.16.

Taguchi's method was the only MDO technique tested that did not find the true optimal solution – it did not even find any of the best solutions in the optimal system architecture family arc. Additionally, the use of precise orthogonal arrays in Taguchi's method tends to force the design vector to be tailored to the test matrix. In other words, the solution technique drives the formulation of the problem rather than vice-versa. For example, in the TPF test study the heliocentric orbital radii of 6.0 AU and 6.5 AU, which were not in the original system trade space, were required to create a valid orthogonal test matrix. Increasing the number of variables and potential variable values in the design vector will only make the formulation of an orthogonal test matrix more difficult, perhaps even become prohibitive for systems with larger, more complex trade spaces. Based on the results of this work, Taguchi's method does not appear to be the best MDO approach for the conceptual design of distributed satellite systems with large trade spaces.

The heuristic simulated annealing algorithm found the optimal solution only once, but consistently found the best solutions with a greater degree of statistical confidence than any of the other MDO techniques. This result is evidenced by the fact that the simulated annealing algorithm recorded the lowest *MSE* of 397.1 (\$K/Image)<sup>2</sup>. This strong showing of simulated annealing is consistent with the application of simulated annealing to other difficult nonconvex problems [Brooks et al, 1998] and is most likely a result of the algorithm's ability to escape local minima by sometimes allowing random moves to neighboring solutions worse than the current solution. A local minima in the TPF single objective trade space is a design vector (i.e. system architecture) that has a lower CPI than any neighboring design vector, but does not have the lowest CPI in the entire system trade



space. As more complex distributed satellite systems with larger design vectors are likely to contain many local minima throughout the trade space, simulated annealing appears to be the most promising MDO technique to apply to the optimization of future DSS conceptual design problems.

Of the four MDO methods explored, the pseudogradient algorithm was the second most likely to converge upon local minima. This entrapment in local minima occurs because the pseudogradient algorithm has no way to extricate itself when all of the investigated local gradients are positive. When this situation occurs, the algorithm believes it has found the optimal solution and becomes permanently trapped in the local minima. The fact that the pseudogradient search found the true optimal solution four out of ten times for TPF can be attributed to the fact that the case study contained a relatively shallow trade space with only 640 possible solutions. Based upon these results, the pseudogradient search algorithm using a modified finite differencing approach does not appear to be a promising technique for optimizing the conceptual design of future distributed satellite systems with larger, deeper trade spaces.

The final MDO approach tested, the univariate search algorithm, also converged upon the true optimal solution four out of ten times, but did so with a lower degree of confidence statistically than the pseudogradient and univariate search algorithms. Unlike the pseudogradient approach, the univariate search algorithm explores every possible neighbor before moving to a new solution. The univariate search algorithm still fails to capture the coupling between design vector parameters, and is prone to getting trapped in local minima. For these reasons, the univariate search algorithm does not appear to be a promising technique for optimizing the conceptual design of future distributed satellite systems.

To summarize, four separate MDO techniques were applied to the single objective optimization of the conceptual design of a complex, highly coupled DSS – the TPF mission. In particular, the simulated annealing algorithm performed best as it efficiently found the best conceptual design solutions with the highest degree of confidence after evaluating

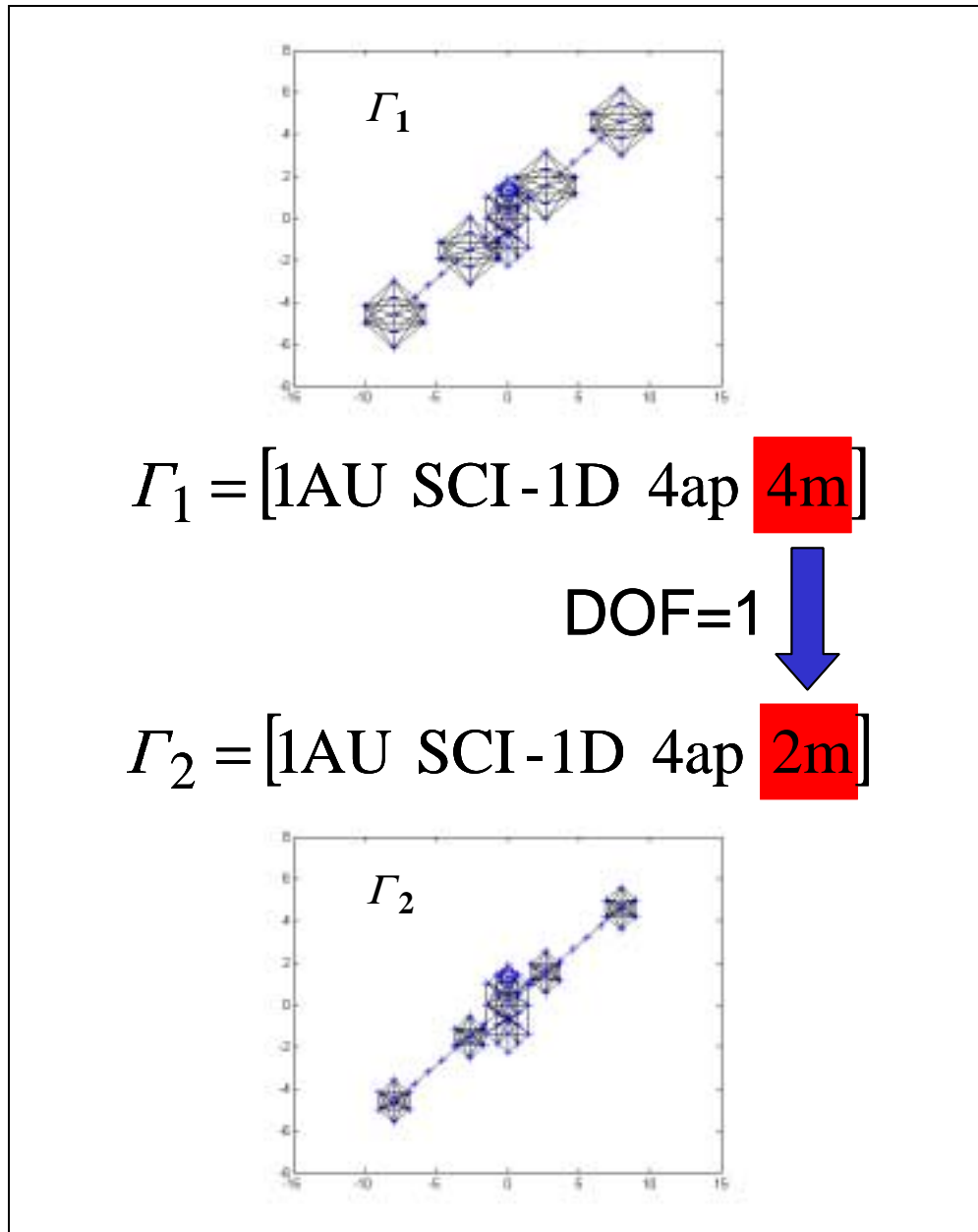
less than eight percent of the full-factorial TPF trade space. Taguchi's method converged on a good solution, but none of the best system architectures in the optimal arc family. Both the pseudogradient and univariate search algorithms frequently became trapped in local minima, and thus performed poorly. The full-factorial data were used to verify the performance of the MDO techniques.

Thus, a survey of different MDO techniques has found that the heuristic simulated annealing algorithm performs the best on the DSS conceptual design single objective optimization problem. This appears to be due to the ability of the simulated annealing algorithm to escape local minima within a nonconvex DSS trade space. As a result of the observations from this experiment, the simulated annealing algorithm forms the core of all the single objective and multiobjective optimization algorithms used in the MMDOSA methodology. The next section investigates how to improve the performance of the single objective optimization simulated annealing algorithm.

### **Improving Single Objective Simulated Annealing Algorithm Performance**

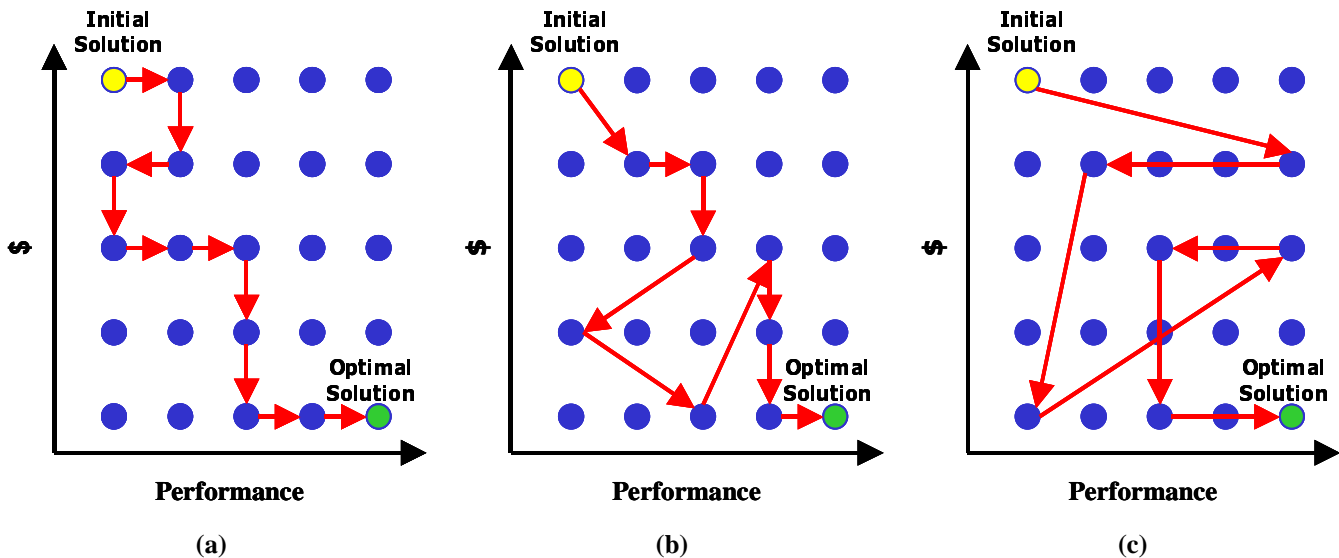
This section builds upon the previous section by searching for ways to improve the observed performance of the simulated annealing algorithm. In particular, the focus will be on selecting the degrees of freedom (DOF) given to the algorithm when moving from one system architecture to another within the trade space. The degrees of freedom is defined as the number of elements (i.e. design variables) within the design vector that are allowed to change between each iteration of the simulated annealing algorithm, and is a parameter the systems engineer has control over when designing the algorithm to explore the DSS trade space. Figure 3.18 illustrates two neighboring TPF design vectors (i.e. system architectures) when the DOF parameter is set to a value of one. The question to be answered is: "What DOF allows simulated annealing to search the single objective DSS trade space most efficiently and why?"

The focus here is on how to perturb the current design vector illustrated in Eqn. 3.23 when moving from one system architecture in the trade space to another. Remembering that the



**Figure 3.18** Neighboring TPF Design Vectors when DOF=1

degrees of freedom is the number of elements within the design vector that are allowed to be perturbed between each iteration of the simulated annealing algorithm, Figure 3.19 illustrates conceptually how the algorithm will search the trade space of a DSS single objective optimization (i.e. minimize CPF) conceptual design problem with a design vector containing five variables as the DOF parameter increases.

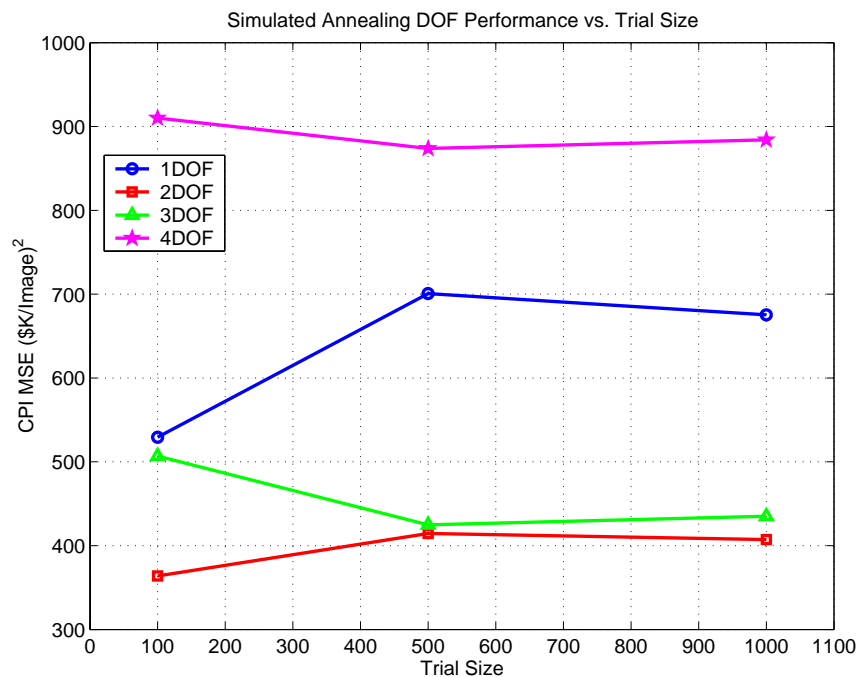


**Figure 3.19** Trade Space Search as a Function of Simulated Annealing DOF: (a) DOF=1, (b) DOF=3, (c) DOF=5

As one can see, when DOF=1, the algorithm is limited to moving within the immediate neighborhood of the current baseline system architecture, where the neighborhood is defined as consisting of all the design vectors which are identical to the original design vector except for a single entry. When the DOF is increased to three, the size of the local neighborhood increases, allowing the algorithm to move a farther distance between iterations. Finally, when the DOF is set equal to the total number of elements in the design vector (i.e. its maximum possible value), the algorithm degenerates into a random search as the local neighborhood is now equivalent to the entire design trade space, meaning that the algorithm may move from any system architecture to any other system architecture within the global trade space between iterations.

To test how the DOF parameter affects the simulated annealing algorithm when searching the trade space of a real DSS, an experiment was carried out on the TPF case study design problem (Eqn. 3.24). The initial system temperature  $T_i$  was set to 1000, and was decreased by a factor of 0.85 after each iteration. The algorithm was terminated after the completion of 48 iterations.

In the experiment, 1000 separate trials of the simulated annealing algorithm were run with the DOF set to each possible value of one, two, three, and four. Figure 3.20 illustrates the results from this experiment. As one can see, the simulated annealing algorithm performed best (i.e. consistently found solutions closest to the true optimal solution) when the DOF parameter was set to a value of two. The worst performance by far was exhibited when DOF was set to its maximum possible value of four, which in this case is equivalent to a nearly random search.



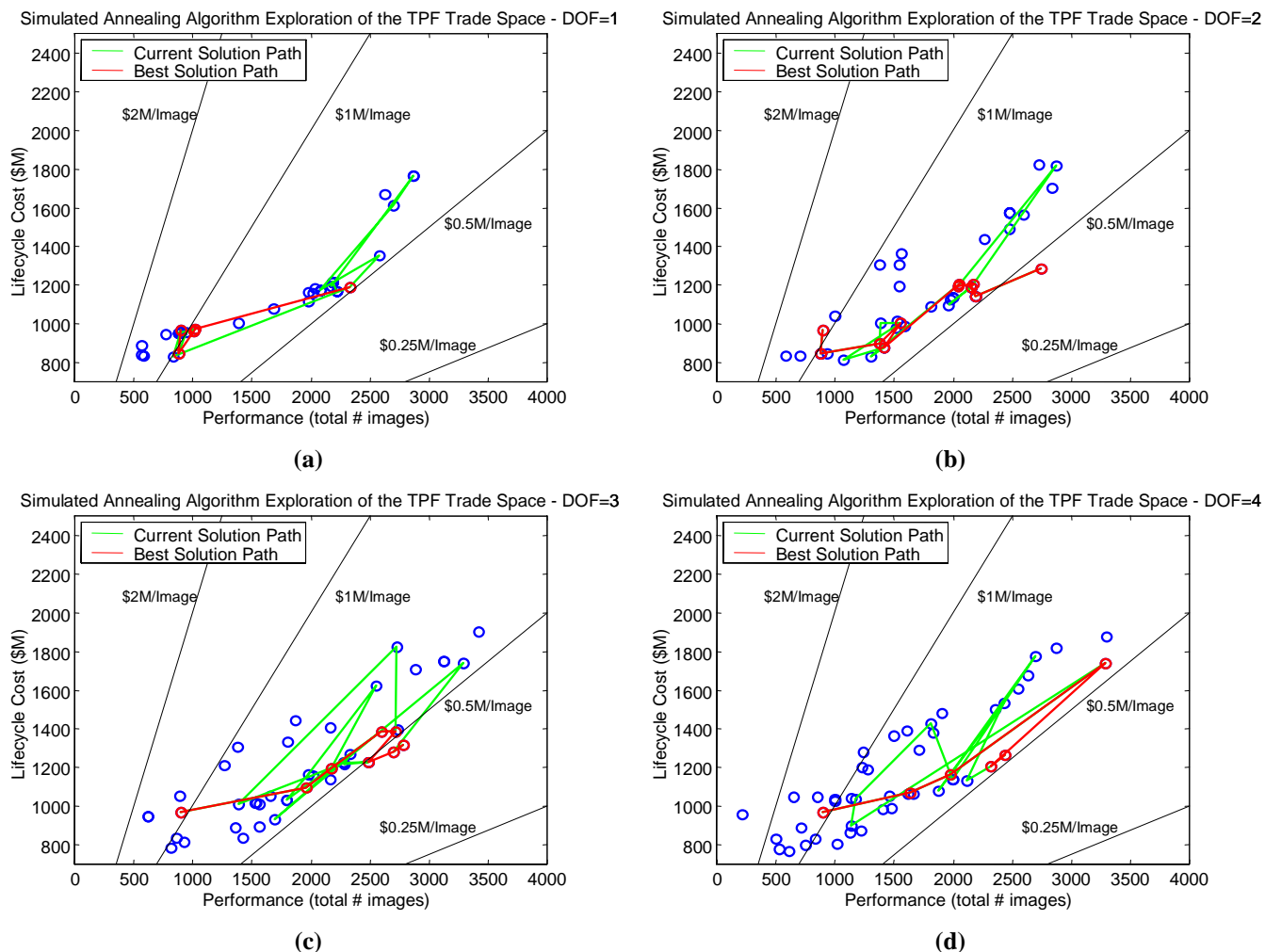
**Figure 3.20** Performance of the Simulated Annealing Algorithm in Finding the Minimum CPI TPF System Architecture as a Function of DOF

Figure 3.21 contains four plots illustrating the exploration of the TPF trade space by a simulated annealing algorithm with a DOF value ranging from one to four. All four trials begin from the same initial solution (i.e. same initial design vector), and are terminated after 48 iterations. The objective function in these trials was to minimize the CPI, and therefore the best designs are located in the lower right corner of these plots. The diagonal

lines represent lines of constant CPI within the trade space. As one can see, even though all four trials begin at the same initial solution, they follow different paths through the TPF trade space and converge upon different final design solutions. When  $\text{DOF}=1$ , the heuristic algorithm takes small steps through the trade space as indicated by the tight clustering of points in the upper left plot of Figure 3.21a. Also notice in the  $\text{DOF}=1$  case that only 26 different TPF design architectures are explored in the algorithm's 48 iterations. Because the design vector is relatively small (i.e. the local neighborhood of solutions is smaller) and only one element of the design vector is perturbed between iterations, several system architectures in the trade space are revisited more than once, contributing to the inefficiency of the algorithm in the  $\text{DOF}=1$  case.

These observations change as the  $\text{DOF}$  value increases in the succeeding plots. First, there is a greater spreading between design points, evidence of the algorithm being able to take bigger jumps between iterations. Second, the total number of design points explored increases and approaches 48, evidence that there is less repetition and therefore the algorithm becomes more efficient. Third, the scatter in the current path of the algorithm indicated by the green line increases, further evidence of the algorithm's ability to jump farther as the  $\text{DOF}$  increases. Finally, as expected from the results illustrated in Figure 3.20, the 2- $\text{DOF}$  and 3- $\text{DOF}$  algorithms converged upon the best solutions ( $\text{CPI}<\$0.5\text{M/Image}$ ), while the 1- $\text{DOF}$  and 4- $\text{DOF}$  algorithms did not perform as well ( $\text{CPI}>\$0.5\text{M/Image}$ ).

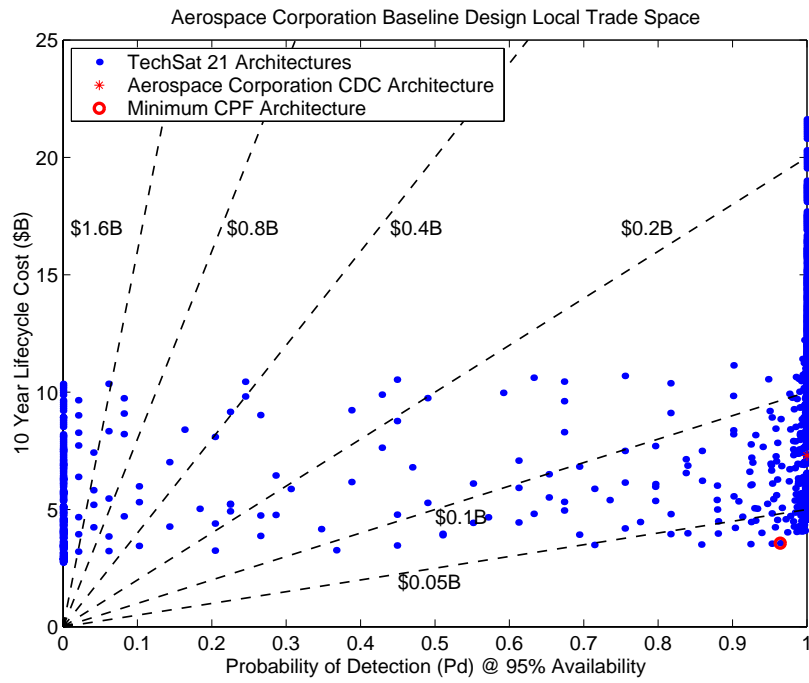
A similar experiment was performed on The Aerospace Corporation Conceptual Design Center (CDC) TechSat 21 baseline design trade space of 1040 architectures. The CDC point design and the CDC trade space are presented and discussed in Chapter 5. Figure 3.22 illustrates the local trade space of The Aerospace Corporation CDC TechSat 21 baseline design as determined from complete enumeration of all 1040 design vectors. Notice that there exists a group of system architectures with a CPF of less than  $\$0.05\text{B}$  per unit probability of detection.



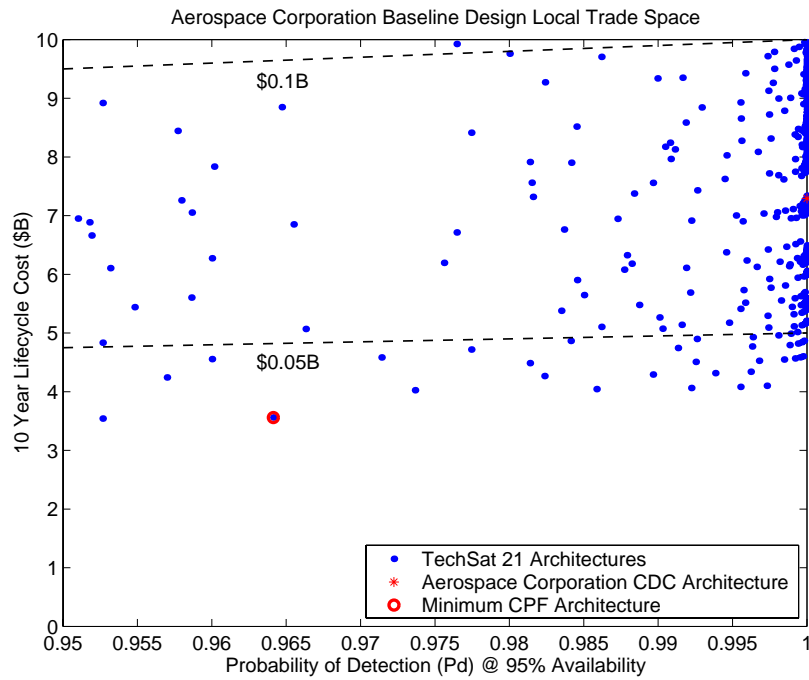
**Figure 3.21** Simulated Annealing Algorithm Exploration of the TPF Trade Space for (a) DOF=1, (b) DOF=2, (c) DOF=3, and (d) DOF=4

In the experiment, multiple trials of the single objective simulated annealing algorithm were run from random starting points with the DOF parameter set to each possible value (i.e. one, two, and three). Each trial began at the same point in the trade space for all three DOF values. The results from this experiment are summarized in Table 3.17.

The single objective simulated annealing algorithm with all three DOF values successfully found members of the family of most cost-effective system architectures – architectures with a CPF less than \$50M. However, the simulated annealing algorithm with DOF=2 performed the best as exhibited by the fact that it exhibited the lowest mean squared error



(a)



(b)

**Figure 3.22** Local Trade Space (a) and Local Trade Space Zoom-In (b) of The Aerospace Corporation CDC TechSat 21 Baseline Design



**TABLE 3.17** Squared Error (SE) and Mean Square Error (MSE) of the Single Objective Simulated Annealing Algorithm as a Function of DOF

Trial	1-DOF CPF (\$M/Pd)	1-DOF SE (\$M/Pd) <sup>2</sup>	2-DOF CPF (\$M/Pd)	2-DOF SE (\$M/Pd) <sup>2</sup>	3-DOF CPF (\$M/Pd)	3-DOF SE (\$M/Pd) <sup>2</sup>
1	43.6	44.9	<b>36.9</b>	0.00	48.0	123.2
2	40.9	16.0	43.4	42.3	40.9	16.0
3	38.1	1.44	45.1	67.2	41.1	17.6
4	38.1	1.44	45.1	67.2	41.3	19.4
5	45.1	67.2	45.7	77.4	41.0	16.8
6	43.4	42.3	43.4	42.3	48.7	139.2
7	38.1	1.44	38.1	1.44	48.4	132.3
8	<b>36.9</b>	0.00	44.3	54.8	45.4	72.3
9	45.1	67.2	40.9	16.0	<b>36.9</b>	0.00
10	<b>36.9</b>	0.00	37.2	0.09	41.1	17.6
11	43.4	42.3	37.2	0.09	38.1	1.44
12	43.6	44.9	<b>36.9</b>	0.00	47.2	106.1
13	40.9	16.0	45.4	72.3	43.6	44.9
14	41.0	16.8	41.1	17.6	41.0	16.8
15	<b>36.9</b>	0.00	41.0	16.8	43.4	42.3
16	<b>36.9</b>	0.00	<b>36.9</b>	0.00	38.1	1.44
17	<b>36.9</b>	0.00	38.1	1.44	43.4	42.3
18	43.4	42.3	<b>36.9</b>	0.00	38.1	1.4
19	37.2	0.09	<b>36.9</b>	0.00	37.2	0.09
20	37.2	0.09	<b>36.9</b>	0.00	41.3	19.4
21	43.6	44.9	37.2	0.09	37.2	0.09
22	41.1	17.6	<b>36.9</b>	0.00	41.1	17.6
23	40.9	16.0	40.9	16.0	41.3	19.4
24	38.1	1.44	40.9	16.0	43.4	42.3
25	38.1	1.44	41.0	16.8	<b>36.9</b>	0.00
26	45.4	72.3	41.0	16.8	42.6	32.5
27	38.1	1.44	41.3	19.4	43.4	42.3
28	<b>36.9</b>	0.00	37.2	0.09	<b>36.9</b>	0.00
29	<b>36.9</b>	0.00	<b>36.9</b>	0.00	38.1	1.44
30	43.4	42.3	37.2	0.09	41.0	16.8
<b>MSE</b>		<b>20.1</b>		<b>18.7</b>		<b>33.4</b>

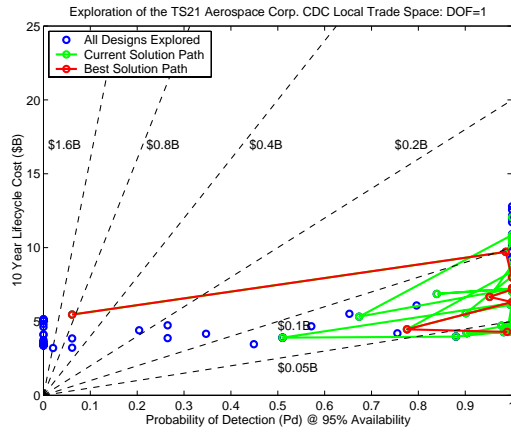
\*True Optimal Solution = **36.9 \$M/Pd**

(Eqn. 3.34). The observation that a 2-DOF simulated annealing algorithm works best concurs with results from the exploration of other DSS trade spaces [Jilla & Miller, 2000; Jilla & Miller, 2001].

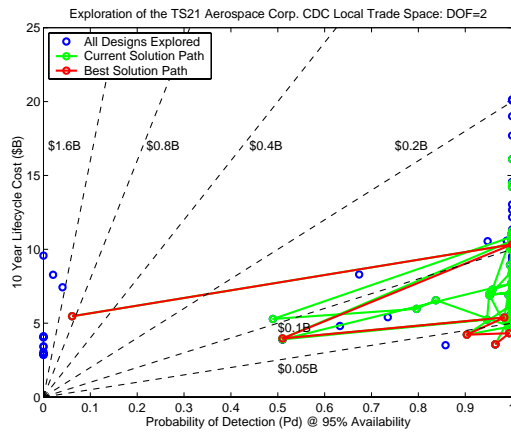
Figure 3.23 contains three plots illustrating the exploration of the TechSat 21 CDC trade space by the single objective simulated annealing algorithm with a DOF value ranging

from one to three. All three trials begin from the same initial architecture, and are terminated after 100 iterations. The objective function in these trials was to minimize the CPF, and therefore the best designs are located in the lower right corner of each plot. The diagonal dashed ISO-CPF lines represent lines of constant CPF within the trade space. As one can see, even though all three trials begin at the same initial solution, they follow different paths through the local trade space and converge upon different final design solutions. While the DOF=1 and DOF=3 algorithms failed to find a design architecture located within the family of most cost-effective design solutions, the DOF=2 algorithm did successfully converge upon an architecture within this family (i.e. an architecture with a CPF below \$40M).

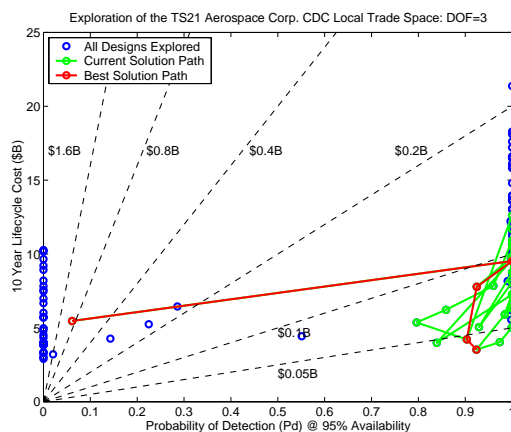
To summarize, in both of these experiments, setting the DOF parameter to a value of two yielded the best single objective optimization simulated annealing algorithm results. This may be due to several reasons. First, when the DOF is set to the minimum value of one, it may be easier to get trapped in a particular region of the trade space as the algorithm takes only small steps between iterations. Conversely, when the DOF is set to the maximum value equivalent to the total number of variables in the design vector, the algorithm essentially engages in a random search and no longer takes advantage of the neighborhood search properties. In other words, the algorithm is more likely to jump to a worse solution in a completely different region of the trade space rather than move to a solution close to the current best solution which is more likely to be better than the current best solution. Thus, setting DOF to a value around two or three yields the best exploration of the DSS trade space as it represents a compromise between the two conflicting factors described above. In conclusion, setting DOF to a value of two appears to provide the most efficient single objective optimization simulated annealing algorithm, and this corresponds with historical studies of similar operations research optimization problems [Bertsimas & Tsitsiklis, 1997].



(a)



(b)



(c)

**Figure 3.23** Single Objective Simulated Annealing Algorithm Exploration of The Aerospace Corporation CDC TechSat 21 Baseline Design Local Trade Space for DOF=1,2,&3

### **Single Objective Optimization Summary**

This section has investigated four separate MDO techniques – Taguchi’s method, heuristic simulated annealing, pseudogradient search, and univariate search – for their effectiveness in finding the optimal solution to the single objective DSS conceptual design problem. The simulated annealing algorithm was found to perform best due to its ability to escape local minima within the nonconvex DSS trade space. Next, methods for improving the performance of the simulated annealing algorithm were explored, and it was found that setting the DOF parameter to a value of two yields the best performance. With the development of this algorithm, a tool now exists to search for the best system architectures on the basis of a single metric within the DSS trade space. It is now time to broaden the DSS conceptual design problem scope to encompass multiobjective optimization problems.

### **3.5.2 Multiobjective Optimization**

The previous section considers only single objective optimization problems. While single objective optimization provides a powerful tool to explore the trade space of a DSS, there are times when several important decision criteria are present simultaneously, and it is not possible or wise to combine these criteria into a single number [Goldberg, 1989]. These problems are classified as multiobjective, or multicriteria, optimization problems – problems in which one attempts to optimize several separate criteria simultaneously. Such problems are important to the DSS conceptual design problem because true systems methods handle trades, not just a single objective function. This is because in real systems engineering problems, one has to balance multiple requirements while trying to achieve multiple goals simultaneously. Therefore any MDO methodology for DSS conceptual design needs to be able to handle multiple objective functions (i.e. multiple decision criteria present simultaneously).

As explained in Chapter 2, several important differences exist between single objective and multiobjective optimization problems. For example, single objective problems contain only one true solution – there exists a single optimal value to minimize or maximize

the objective function. Multiobjective problems, however, can have several valid solutions in the Pareto optimal set. Section 2.2.5 reviews these differences in greater detail and also presents the formal structure of a multiobjective optimization problem.

### Searching for a Single Architecture in the Pareto Optimal Set

The first foray into applying multiobjective optimization to the conceptual design of distributed satellite systems will involve searching for a single system architecture in the Pareto optimal set. Recall that Eqn. 3.24 gives the mathematical formulation for the single objective optimization instance of the TPF conceptual design problem. For the multiobjective case, this formulation must be rewritten as:

$$\begin{aligned}
 \text{Objective:} & \quad \text{Min} \left( \sum_{y=1}^5 \Phi_y(\Gamma) \right) \text{ AND } \text{Max} \left( \sum_{y=1}^5 \Psi_y(\Gamma) \right) \\
 \text{Constraints:} & \quad \textit{Subject to} \\
 \text{Isolation} & \quad \theta_r \geq 20 \text{ milliarc sec} \\
 & \quad \Omega \leq 10^{-6} \\
 \text{Integrity} & \quad \text{SNR}_s \geq 5 \\
 & \quad \text{SNR}_{ms} \geq 10 \\
 & \quad \text{SNR}_{ds} \geq 25 \\
 \text{Availability} & \quad A_v \geq 0.95 \\
 \text{Bounds} & \quad 1.0\text{AU} \leq \gamma_1 \leq 5.5 \text{ AU} \\
 & \quad 4 \leq \gamma_3 \leq 10 \\
 & \quad 1 \text{ m} \leq \gamma_4 \leq 4 \text{ m}
 \end{aligned} \tag{3.35}$$

In addition to the change in the problem formulation, the multiobjective approach also requires a change in the internal mechanics of the simulated annealing algorithm. Eqn. 3.31 states the  $\Delta$  parameter that is used to determine the probability with which the simulated annealing algorithm may move to a worse solution with the intent of escaping local optima within the trade space. In the single objective case,  $\Delta$  is simply the difference in

the objective function value between successive steps of the algorithm. For multiobjective problems, however,  $\Delta$  must take into account the difference in successive steps between all objective functions – that is for all  $N$  dimensions of the problem (i.e. for all decision criteria). Further complicating matters is the fact that the units for each objective function will likely be different, making it difficult for the  $\Delta$  parameter to properly compare "apples to apples."

To solve this problem, a new approach has been developed to compute the  $\Delta$  parameter for multiobjective problems. The steps in this approach are listed below.

1. Take a random sample of the DSS trade space.
2. Compute the maximum range  $\delta$ ,  $1\sigma$  range,  $2\sigma$  range, and  $3\sigma$  range of each objective function within the random sample.
3. Normalize the observed difference between successive steps of the algorithm for each objective function by the maximum range  $\delta$ ,  $1\sigma$  range,  $2\sigma$  range, or  $3\sigma$  range for that objective function as computed in step two.
4. Set  $\Delta$  equal to the sum of the normalized values computed in step three divided by the total number  $N$  dimensions in the problem.

The formula for computing the  $\Delta$  parameter using the above steps is:

$$\Delta = \frac{\sum_{n=1}^N \frac{E_n(\Gamma_{i+1}) - E_n(\Gamma_i)}{\delta_n}}{N} \quad (3.36)$$

where  $N$  is the total number of dimensions (i.e. objective functions or decision criteria) in the problem,  $i$  is the iteration number within the simulated annealing algorithm, and  $\delta_n$  is the maximum range in dimension  $n$  observed in the random sample. Depending upon the distribution of the random sample,  $\delta_n$  may be replaced by the  $1\sigma_n$  range,  $2\sigma_n$  range, or  $3\sigma_n$  range for each dimension  $n$  in the objective function.

Using the information learned from the random sample of the TPF trade space, the simulated annealing algorithm  $\Delta$  parameter in Eqn. 3.36 may now be specifically computed for the TPF multiobjective optimization problem in Eqn. 3.35 as:

$$\Delta = \frac{\sum_{y=1}^5 \Phi_y(\Gamma_{i+1}) - \sum_{y=1}^5 \Phi_y(\Gamma_i)}{\$1394 \times 10^6} + \frac{\sum_{y=1}^5 \Psi_y(\Gamma_i) - \sum_{y=1}^5 \Psi_y(\Gamma_{i+1})}{3640 \text{ images}} \quad (3.37)$$

where  $y$  is the year in the mission. Notice the difference in Eqn. 3.37 in the order of the new design architecture  $\Gamma_{i+1}$  and previous design architecture  $\Gamma_i$  between the lifecycle cost objective  $\Phi$  and the target probability of detection objective  $\Psi$ . Remember that the  $\Delta$  parameter is used to compute the probability that the simulated annealing algorithm will move to a worse solution in the trade space with the intention of avoiding entrapment within a local optima (i.e. a local minima when one is attempting to minimize a particular metric and a local maxima when one is attempting to maximize a particular metric). This probability should be higher when the new objective function value is closer to the current objective function value than when the new objective function value is further from the current objective function value. A worse solution means a larger objective function value for a minimization objective and smaller objective function value for a maximization objective. This accounts for the observed difference in order in Eqn. 3.37 between the lifecycle cost (i.e. a minimization objective) and target probability of detection (i.e. a maximization objective).

Once the parameter has been computed, the decision logic for moving from one current baseline solution to a new current baseline solution within the simulated annealing algorithm must be defined. For the original single objective minimization problem, this logic is:

$$\begin{aligned} &\text{IF } E(\Gamma_{i+1}) < E(\Gamma_i) \text{ OR } \chi < e^{-\frac{\Delta}{T}} \\ &\quad \Gamma_b = \Gamma_{i+1} \\ &\text{ELSE} \\ &\quad \Gamma_b = \Gamma_i \\ &\text{END} \end{aligned} \quad (3.38)$$

where  $E(\Gamma_{i+1})$  is the CPF of the new design vector,  $i$  is the iteration number within the simulated annealing algorithm,  $\chi$  is a value created by a random number generator,  $\Delta$  is the mean normalized difference in the objective function value as computed in Eqn. 3.36,  $T$  is the system temperature, and  $\Gamma_b$  is the current baseline design vector.

For multiobjective problems, the decision logic when searching for a single point in the Pareto optimal set must be changed to:

$$\begin{aligned}
 &\text{IF } E_n(\Gamma_{i+1}) < E_n(\Gamma_i) \text{ for ALL } n = 1, 2, \dots, N \text{ OR } \chi < e^{-\frac{\Delta}{T}} \\
 &\quad \Gamma_b = \Gamma_{i+1} \\
 &\text{ELSE} \\
 &\quad \Gamma_b = \Gamma_i \\
 &\text{END}
 \end{aligned} \tag{3.39}$$

where  $E_n(\Gamma_{i+1})$  is the objective value along dimension  $n$  of the  $N$ -dimensional problem. The new architecture is accepted by the simulated annealing algorithm as the new current baseline architecture if the new architecture rates better for all decision criteria than the current baseline architecture or randomly. For the multiobjective TPF example in Eqn. 3.35, this translates into an architecture which has both a lower lifecycle cost and a greater performance (i.e. produces more images) than the baseline architecture. This logic will be referred to hereafter as the multiobjective single solution, or single point, simulated annealing algorithm.

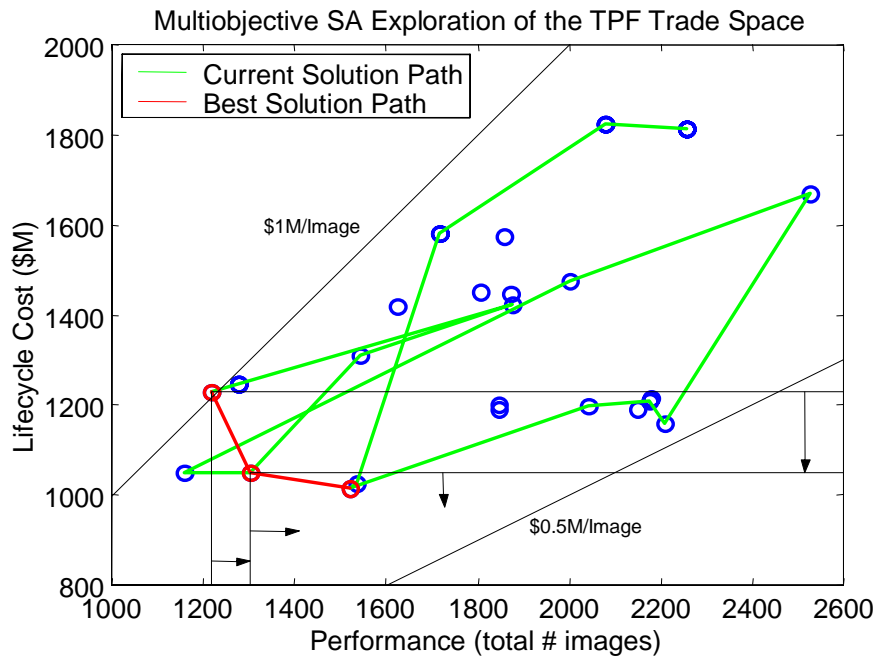
The decision logic in Eqn. 3.39 holds only when the goal is to minimize all  $N$  decision criteria. In practice, any maximization problem may be rewritten as a minimization problem by multiplying the objective function by negative one. Using the predefined nomenclature for the TPF case study, the decision logic in Eqn. 3.39 may be written for the TPF multiobjective optimization problem (Eqn. 3.35) explicitly as:



$$\begin{aligned}
 &\text{IF } \sum_{y=1}^5 \Phi_y(\Gamma_{i+1}) < \sum_{y=1}^5 \Phi_y(\Gamma_i) \quad \text{AND} \quad \sum_{y=1}^5 \Psi_y(\Gamma_{i+1}) > \sum_{y=1}^5 \Psi_y(\Gamma_i) \quad \text{OR} \quad \chi < e^{-\frac{\Delta}{T}} \\
 &\quad \Gamma_b = \Gamma_{i+1} \\
 &\text{ELSE} \\
 &\quad \Gamma_b = \Gamma_i \\
 &\text{END}
 \end{aligned} \tag{3.40}$$

Figure 3.24 illustrates how the multiobjective simulated annealing algorithm searches the TPF trade space with this decision logic. In the figure, a zoom-in of the region within the global TPF trade space where the algorithm has traversed is shown. The green line plots the path of the current baseline solution  $\Gamma_b$  in Eqn. 3.40. The scatter in the path of the green line to worse solutions is due to the random moves (i.e. third inequality in Eqn. 3.40) of the algorithm. The red line illustrates the path of the best solution (i.e. the anticipated Pareto optimal system architecture). The vertical and horizontal black lines illustrate the bounds the algorithm places on the trade space upon moving to each best solution (i.e. the first and second inequalities in Eqn. 3.40). These bounds constrain the algorithm to move towards the design architectures located in the lower right corner of the TPF trade space as illustrated by the arrows emanating from the bounds.

It is important to remember that the path of the multiobjective simulated annealing algorithm will depend on the initial starting point. This is because the randomly selected initial starting point determines the first set of bounds placed on the trade space, and, using the current decision logic in Eqn. 3.39 and Eqn. 3.40, these bounds automatically eliminate a large portion of the conceptual design trade space from consideration. Included in the eliminated area of the trade space may be design architectures which fall within the Pareto optimal set. However, Pareto optimal architectures will also exist within the original bounds unless the initial starting point is also Pareto optimal, and the algorithm should converge upon one of these Pareto optimal architectures.



**Figure 3.24** Multiobjective Single Solution Simulated Annealing Search of the TPF Trade Space Zoom-In ( $i=48$  iterations)

Notice the difference between how the new multiobjective (Figure 3.24) and original single objective (Figure 3.21) simulated annealing algorithms search the trade space. The single objective algorithm always aims for the *same* point (i.e. the minimum CPI architecture) in the Pareto optimal set, while the multiobjective algorithm searches for *any* point in the Pareto optimal set. Thus, several runs of the multiobjective single solution simulated annealing algorithm should yield either a portion of or even the entire Pareto optimal set, or boundary/front for a two-dimensional multiobjective optimization problem, of the DSS trade space. With the development of this new simulated annealing algorithm, there now exists a method for handling multiobjective DSS design problems.

### Searching for Multiple Architectures in the Pareto Optimal Set Simultaneously

The multiobjective algorithm described in the previous section finds only a single architecture in the Pareto optimal set during each run. In fact, the continual application of bounds on each objective function prevents the algorithm from finding many of the system architectures in the Pareto optimal set, depending upon the location of the random ini-

tial starting point. It might be more efficient to search simultaneously for multiple points in the Pareto optimal set during the course of a single run of the simulated annealing algorithm. This will be the focus of the new simulated annealing algorithm developed here, referred to hereafter as the multiobjective multiple solution, or multiple point, simulated annealing algorithm.

To find multiple Pareto optimal design architectures at once, the simulated annealing algorithm derived in the previous section will need to be modified in two major ways:

1. The decision logic in Eqn. 3.39 for changing the current baseline solution must accommodate moving in *any* direction towards and along the Pareto boundary/front.
2. All past design points traversed by the algorithm must be kept in memory so that every new system architecture may be checked against the past design points to compute the continually evolving Pareto optimal set.

For the TPF example, the problem formulation remains the same as stated in Eqn. 3.35. During each step of the multiple solution multiobjective simulated annealing algorithm, the newest design architecture is checked against the following decision logic – assuming the objective is to minimize each decision criteria – to determine whether or not the new design architecture is a candidate for Pareto optimality:

$$\begin{aligned}
 & \text{IF } E_n(\Gamma_k) < E_n(\Gamma_i) \text{ for ALL } n = 1, 2, \dots, N \text{ and } k = 1, 2, \dots, i-1 \\
 & \quad \Gamma_i \notin P \\
 & \text{ELSE} \\
 & \quad \Gamma_i \in P \\
 & \text{END}
 \end{aligned} \tag{3.41}$$

where  $\Gamma_k$  represents *all* the previous points (i.e. system architectures) traversed by the algorithm,  $i$  is the current iteration number within the simulated annealing algorithm,  $N$  is the number of dimensions (i.e. number of decision criteria, design goals, or objective functions) in the design problem, and  $P$  is the candidate set of Pareto optimal architectures. Eqn. 3.41 states that if the new design architecture  $\Gamma_i$  is dominated by any of the existing design architectures  $\Gamma_k$ , then the new architecture is not a candidate for Pareto

optimality. If, however, the new design architecture  $\Gamma_i$  is not dominated by any of the other design architectures  $\Gamma_k$  that have been searched up to iteration  $i$  within the algorithm, then it is a candidate for Pareto optimality.

Once all of the iterations within the simulated annealing algorithm have been completed, the true Pareto optimal set  $P^*$  is extracted from the candidate Pareto optimal set  $P$  in the following manner:

```

FOR  $k = 1:i$ 
  FOR  $z = 1:i$ 
    IF  $E_n(\Gamma_z) < E(\Gamma_k)$  for ALL  $n = 1, 2, \dots, N$ 
       $\Gamma_k \notin P^*$ 
    ELSE
       $\Gamma_k \in P^*$ 
    END
  END
END
END

```

(3.42)

where  $i$  is the size (i.e. the total number of system architectures) of  $P$ . Eqn. 3.41 and Eqn. 3.42 detail how the Pareto optimal set of system architectures is extracted during a single run of the multiobjective multiple solution simulated annealing algorithm. The actual path the simulated annealing algorithm takes through the trade space from one current baseline architecture  $\Gamma_b$  to another current baseline architecture still incorporates randomness to prevent entrapment in local minima and follows the decision logic:

```

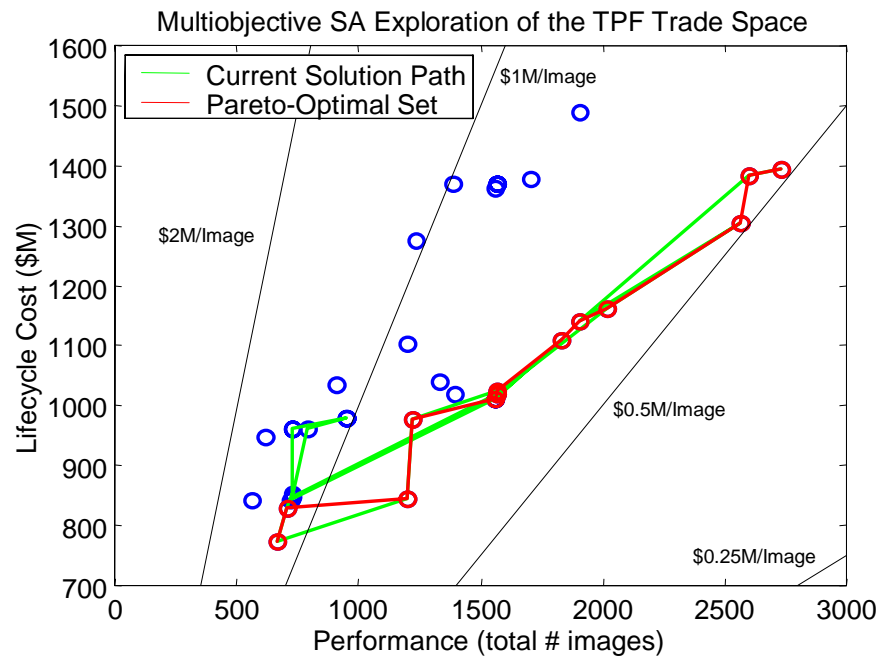
IF  $\Gamma_{i+1} \in P$  OR  $\chi < e^{-\frac{\Delta}{T}}$ 
   $\Gamma_b = \Gamma_{i+1}$ 
ELSE
   $\Gamma_b = \Gamma_i$ 
END

```

(3.43)

where  $i$  is the current iteration number within the simulated annealing algorithm.

Figure 3.25 illustrates how the multiobjective multiple solution simulated annealing algorithm searches the TPF trade space with the new decision logic outlined in Eqn. 3.41 through Eqn. 3.43. This figure shows a zoom-in of the region within the TPF trade space where the algorithm has traversed. The green line shows the path of the current baseline solution  $\Gamma_b$  in Eqn. 3.43. Blue points represent system architectures that are dominated by one or more other system architectures. Red points represent the system architectures that comprise the Pareto optimal set  $P^*$ . The red line connecting these points defines the Pareto optimal boundary, also known as the Pareto optimal front, of this explored subset of the TPF trade space. Along this boundary, the systems engineer cannot improve the performance of the conceptual design without also increasing lifecycle cost. Conversely along this same boundary, the systems engineer cannot decrease lifecycle cost without also harming system performance. Thus, the Pareto boundary successfully captures the *trades* between all of the DSS design decision criteria. In this particular case for TPF, the trade between system performance and lifecycle cost is *quantitatively* captured.



**Figure 3.25** Multiobjective Multiple Solution Simulated Annealing Search of the TPF Trade Space Zoom-In ( $i=48$  iterations)

As before, the path of this new multiobjective multiple solution simulated annealing algorithm will depend on the initial starting point. However, unlike the previous multiobjective single solution simulated annealing algorithm, this algorithm does not use bounds to eliminate different portions of the trade space from consideration. This in turn enables the algorithm to find many Pareto optimal system architectures during a single run as illustrated in Figure 3.25. In fact, the efficiency of this new approach is illustrated by the close concurrence between the current solution (i.e. the green line) and best solution (i.e. the red line) paths, especially when compared with the concurrence, or lack thereof, in the prior two simulated annealing algorithms illustrated in Figure 3.21 and Figure 3.24.

Two separate heuristic simulated annealing algorithms have now been developed to search for Pareto optimal system architectures within the conceptual design trade space of any DSS. But how does the systems engineer know how good the answer for the true Pareto optimal set  $P^*$  is? This will be the subject of the next section.

### **Approximating the True Global Pareto Boundary with an Estimated Pareto Boundary**

In this section, a procedure is developed to evaluate how well the two new multiobjective algorithms – the multiobjective single solution simulated annealing algorithm and the multiobjective multiple solution simulated annealing algorithm – approximate the DSS trade space global Pareto boundary after evaluating only a fraction of the global trade space. This is an important issue because the global Pareto boundary provides the systems engineer with much useful information and insight into the DSS conceptual design trade space. Keeping with the TPF case study example with the simultaneous multiple objectives of minimizing lifecycle cost and maximizing lifecycle performance (Eqn. 3.35), knowledge of the Pareto boundary enables the systems engineer to quickly determine with a high degree of confidence whether or not one can expect to achieve a given level of performance within a given budget (ex: For \$1 billion, can the system image 1000 separate star systems?).

To assess how well an *estimated Pareto boundary* (i.e. the Pareto boundary developed from looking at only a portion of the trade space) approximates the *true global Pareto boundary* (i.e. the Pareto boundary developed from looking at the entire trade space), a mean error (*ME*) metric will be used. The process involves the following steps:

1. Compute the true global Pareto optimal set by completely enumerating the TPF conceptual design trade space.
2. Using the least squares fit technique, fit a curve/equation to this set of points to create the true global Pareto boundary.
3. Compute the Pareto optimal architectures from the subset of the trade space searched by an MDO algorithm.
4. Fit a curve/equation to this Pareto optimal subset to approximate the estimated Pareto boundary.
5. Compute the mean error between the estimated Pareto boundary and the true global Pareto boundary as a metric of how well the estimated boundary approximates the true global boundary.

The remainder of this section discusses each of these steps in detail within the context of the TPF case study.

First, the entire TPF full-factorial trade space must be enumerated, and the set of non-dominated solutions (i.e. nondominated system architectures) extracted to form the Pareto optimal set. Next, a polynomial curve is fit to the Pareto optimal set of points to create the global Pareto boundary. The polynomial curve takes the form of the equation:

$$\hat{y} = a_0 + a_1x + a_2x^2 + \dots + a_nx^n \quad (3.44)$$

where  $n$  is the order of the polynomial,  $\hat{y}$  is the dependent variable,  $x$  is the independent variable, and  $a_0 \dots a_n$  are unknown coefficients. The unknown coefficients can be computed by executing a least squares fit, which minimizes the sum of the squares of the deviations of the data (i.e. the actual Pareto optimal points) from the model (i.e. the polynomial equation). The least squares criterion may be mathematically represented as:

$$\min \sum (y_i - \hat{y}_i)^2 \quad (3.45)$$

where  $y_i$  is the observed value (i.e. the actual Pareto optimal point) of the dependent variable – in this case lifecycle cost – for the  $i$ th observation and  $\hat{y}_i$  is the estimated value from the polynomial equation of the dependent variable for the  $i$ th observation. For an  $n$ th order polynomial, there are  $k$  equations with  $n+1$  unknowns:

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_i \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ 1 & x_3 & x_3^2 & \dots & x_3^n \\ 1 & x_4 & x_4^2 & \dots & x_4^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} \quad (3.46)$$

To solve this set of equations, the number of Pareto optimal points  $i$  must be greater than the order  $n$  of the polynomial (i.e. must have more equations than unknowns).

Figure 3.26 illustrates the global Pareto boundary for the TPF case study. Blue dots represent dominated architectures within the global trade space. Red dots represent nondominated architectures that comprise the Pareto optimal set. The green line represents the least squares fourth order polynomial fit through the nondominated architectures to create the global Pareto optimal boundary along which one cannot improve one system performance without increasing lifecycle cost. This curve is represented by the following fourth-order polynomial equation with a least squares fit of \$42.2M:

$$\hat{y} = -1.93e^{-11}x^4 + 1.47e^{-7}x^3 - 2.4e^{-4}x^2 + 0.208x + 688 \quad (3.47)$$

where  $\hat{y}$  is lifecycle cost and  $x$  is system performance (i.e. total number of images).

When an optimization algorithm then searches a subset of the TPF trade space, an estimated Pareto boundary passing through the nondominated solutions in the search subset is computed in a similar manner via Eqn. 3.44 through Eqn. 3.46. To assess how well the local Pareto boundary approximates the global Pareto boundary, the mean error (ME) may be computed between the two curves:



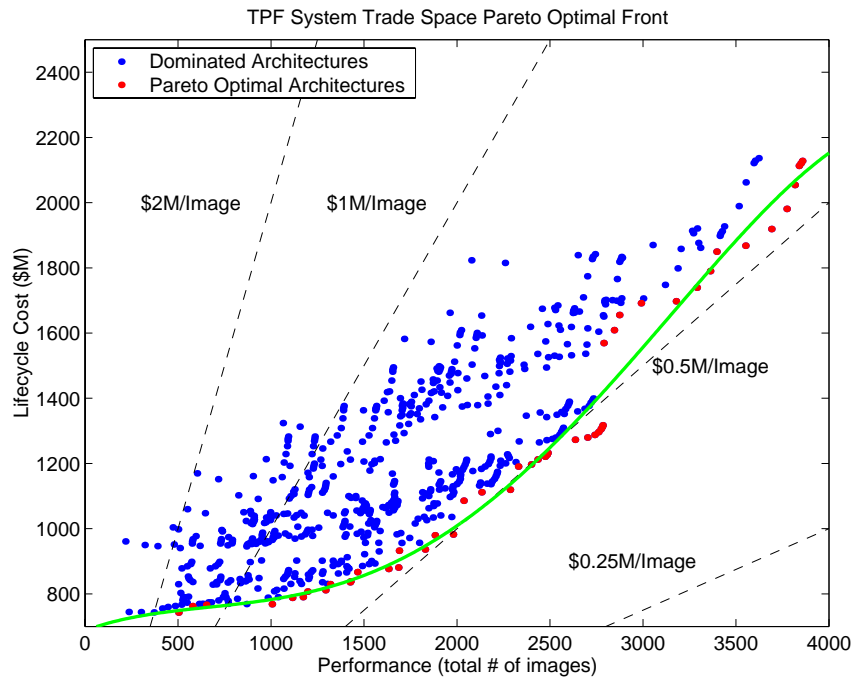


Figure 3.26 TPF Trade Space Global Pareto Boundary

$$ME = \frac{\sum_{x=i_{\min}}^{i_{\max}} \sqrt{(y_x - \hat{y}_x)^2}}{k} \tag{3.48}$$

where  $y_x$  is the  $y$  value of the estimated Pareto curve for a given value of  $x$ ,  $\hat{y}_x$  is the  $y$  value of the true global Pareto curve for the same value of  $x$ ,  $i_{\max}$  minus  $i_{\min}$  represents the span of the curve for which the approximation is evaluated, and  $k$  is the number of points compared between the two curves.

The above approach has been applied to approximate the true global Pareto boundary via an estimated Pareto boundary in the TPF case study. The three following simulated annealing algorithms were tested to determine which one best finds the required nondominated solutions to properly approximate the true global Pareto boundary.

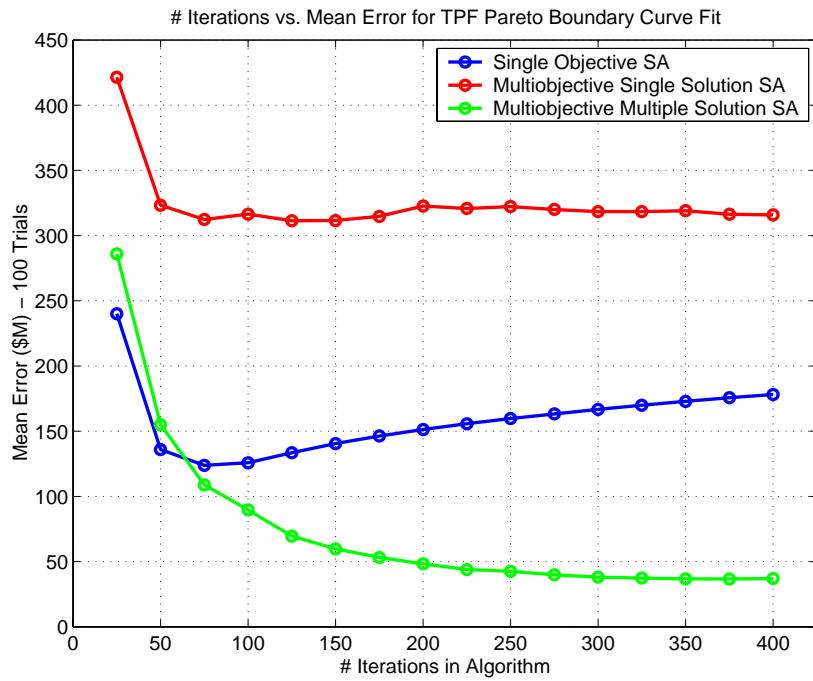
1. Single Objective (Minimize CPI) Simulated Annealing Algorithm (Eqn. 3.23, Eqn. 3.24, Eqn. 3.29 through Eqn. 3.31, and Eqn. 3.38)

2. Multiobjective (Minimize Lifecycle Cost AND Maximize Performance)  
Single Solution Simulated Annealing Algorithm (Eqn. 3.35, Eqn. 3.39, and Eqn. 3.40)
3. Multiobjective (Minimize Lifecycle Cost AND Maximize Performance)  
Multiple Solution Simulated Annealing Algorithm (Eqn. 3.35 and Eqn. 3.41 through Eqn. 3.43)

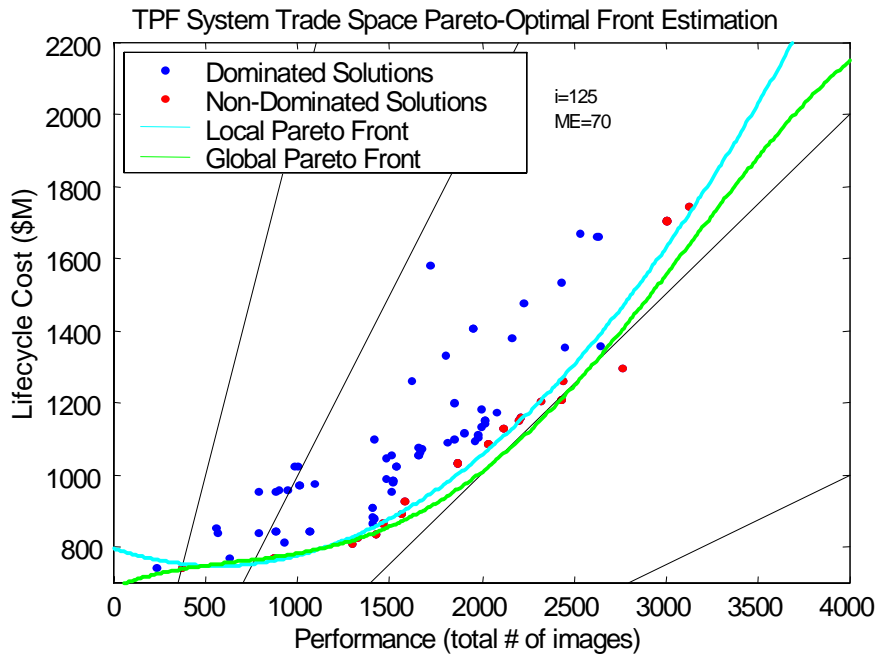
These three algorithms were placed in a test matrix in which each algorithm was applied 100 times from 100 different randomly selected initial starting points to the TPF conceptual design problem and allowed from 25 to 400 iterations in increments of 25. The results from the test matrix appear in Figure 3.27a.

Several interesting observations may be drawn from this figure. First, the multiobjective single solution simulated annealing algorithm appears to provide the worst global Pareto boundary approximation. This is due to the fact that this algorithm searches for only a single architecture in the Pareto optimal set, making it susceptible to concentrating on only a small portion of the global trade space. Thus, the estimated Pareto boundary will closely approximate the true global Pareto boundary in this region, but will poorly approximate the true Pareto boundary in the remaining regions of the global trade space, resulting in the poor ME values.

The single objective simulated annealing algorithm provides the best approximation of the original three algorithms tested when 50 or fewer iterations are executed, but then worsens as the total number of iterations increases. This worsening is an interesting phenomenon that may be attributed to the mechanics of the single objective simulated annealing algorithm. This algorithm always searches for the same point (i.e. the minimum CPF design) and thus always concentrates in this region of the trade space. As the number of iterations increases, the algorithm is allowed to investigate more neighboring design solutions, increasing the bias in this region of the trade space. Thus, the estimated Pareto boundary closely approximates the global boundary in the minimum CPF region of the trade space, but poorly approximates the Pareto boundary everywhere else. This is why the single objective simulated annealing algorithm performs worse as the number of iterations



(a)



(b)

**Figure 3.27** Mean Error Between the Local and Global Pareto Boundary as a Function of the Number of Iterations in Each Algorithm (a) and Approximation of the Global Pareto Boundary by the Multiobjective Multiple Solution Simulated Annealing Algorithm (b)

within the algorithm increases – the bias against the non-minimum CPF regions of the trade space increases.

Beyond 75 iterations, the multiobjective multiple solution simulated annealing algorithm performs best. This is to be expected as this algorithm was specifically created, designed, and tailored to search for the entire set of Pareto optimal architectures. When this multiobjective heuristic algorithm hits the Pareto boundary, it fans out in both directions along the boundary. For this reason, the multiple solution algorithm does not develop the same bias for a particular region of the trade space that the former two algorithms do, leading to better performance in approximating the true global Pareto boundary. Based on these results for the TPF case study, the multiobjective multiple solution simulated annealing algorithm best approximates the global Pareto boundary of the three heuristic simulated annealing algorithms developed in Section 3.5 to search the DSS conceptual design trade space.

Next, an experiment was carried out to test the effectiveness of the various DOF multiobjective multiple solution simulated annealing algorithms in terms of how well each algorithm approximates the TechSat 21 Aerospace Corporation CDC trade space Pareto boundary after evaluating only a fraction of the CDC trade space. To assess the effectiveness of each of the three variants of the core algorithm, the same mean error (Eqn. 3.48) metric was used. The same experimental procedure used to evaluate and compare the effectiveness of the single objective simulated annealing algorithm, the multiobjective single solution simulated annealing algorithm, and the multiobjective multiple solution algorithm in finding the TPF mission Pareto boundary in Section 3.5.2 was used to compare the 1-DOF, 2-DOF, and 3-DOF algorithms:

1. Compute the true global Pareto optimal set by completely enumerating the TPF conceptual design trade space.
2. Using the least squares fit technique, fit a curve/equation to this set of points to create the true global Pareto boundary.
3. Compute the Pareto optimal architectures from the subset of the trade space searched by an MDO algorithm.

4. Fit a curve/equation to this Pareto optimal subset to approximate the estimated Pareto boundary.
5. Compute the mean error between the estimated Pareto boundary and the true global Pareto boundary as a metric of how well the estimated boundary approximates the true global boundary.

In The Aerospace Corporation baseline design trade space, the Pareto boundary curve fit is biased by the abundance of system architectures at the extreme ends of the performance axis in Figure 3.22a. For this reason, only the Pareto optimal system architectures with a target probability detection between  $0.01 < P(d) < 0.99$  are used to develop the least squares curve fit. Under this constraint, the curve approximating the true Pareto boundary within the trade may be represented by the following first order polynomial equation with a ME fit of \$0.127B:

$$\hat{y} = 0.88x + 3.3 \quad (3.49)$$

where  $\hat{y}$  is the lifecycle cost value of the true Pareto boundary and  $x$  is the probability of detection along the true Pareto boundary.

When the multiobjective multiple solution simulated annealing algorithm searches a subset of the TechSat 21 CDC trade space, an approximated Pareto boundary passing through the nondominated solutions in the search subset is computed via Eqn. 3.44 through Eqn. 3.46. To assess how well the approximated Pareto boundary matches the true Pareto boundary, the mean error (Eqn. 3.48) may be computed between the two curves.

This approach has been applied to approximate the Pareto boundary of The Aerospace Corporation CDC baseline design CDC trade space without complete enumeration of the CDC trade space. The 1-DOF, 2-DOF, and 3-DOF versions of the multiobjective simulated annealing algorithm were tested to determine which one best finds the required non-dominated solutions to properly approximate the true Pareto boundary. These three algorithms were placed in a test matrix in which each algorithm was applied multiple times from the same randomly selected initial starting points for each trial within The Aerospace Corporation CDC baseline design trade space. Table 3.18 summarizes the

results in terms of the mean error between the Pareto boundary approximation and the true Pareto boundary. As one can see, the 2-DOF algorithm performed better than the 1-DOF and 3-DOF algorithms. Figure 3.28 illustrates graphically how well each of the algorithms approximated the true Pareto boundary from one trial of this experiment.

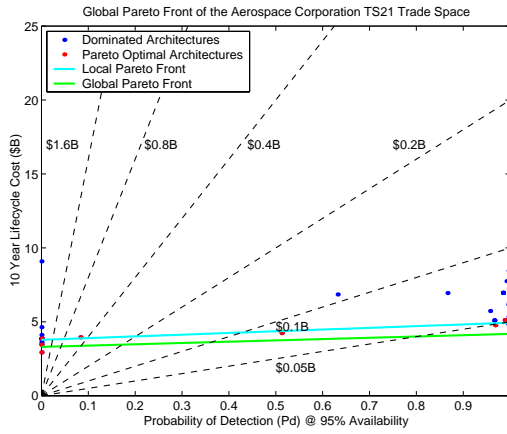
**TABLE 3.18** ME Between the Approximated and True Pareto Boundary for Each Multiobjective Algorithm Variant

Algorithm Variant	ME (\$B)
1-DOF	1.55
2-DOF	0.74
3-DOF	0.87

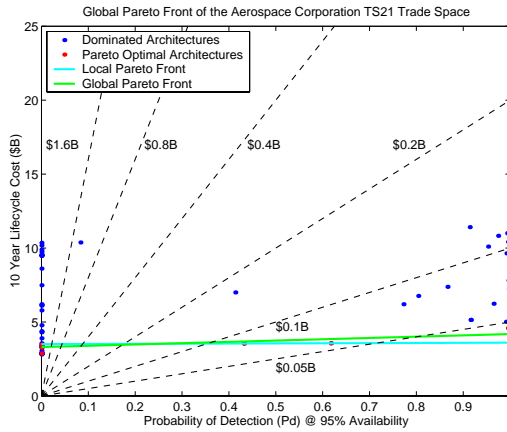
This section has shown that the global Pareto boundary of a DSS trade space may be successfully approximated after evaluating only a fraction of the complete trade space. In particular, the 2-DOF multiobjective multiple solution simulated annealing algorithm provides the best results. With this algorithm, a design tool that enables the systems engineer to approximate the Pareto boundary of any DSS conceptual design trade space now exists. Knowledge of this boundary allows the systems engineer to capture trades between any desired design parameters along the most efficient set of design solutions. For example, in the TPF case study the Pareto boundary tells the systems engineer the maximum performance that can be expected for a given budget (i.e. lifecycle cost) and how this performance will change as the budget increases or decreases. The ability to approximate the global Pareto boundary of a DSS trade space is an important advancement that greatly strengthens the MMDOSA methodology for the conceptual design of distributed satellite systems.

### Higher Order Pareto Optimal Sets

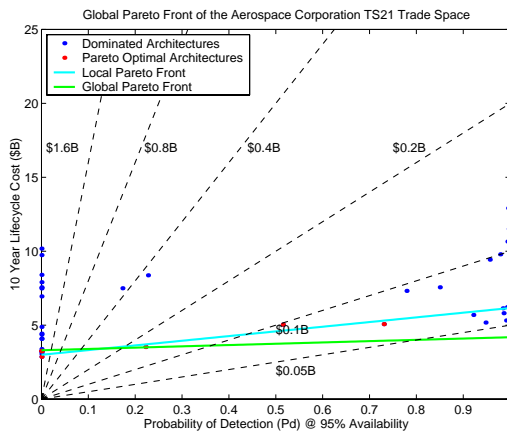
The Pareto optimal set of architectures becomes harder to visualize as the number of dimensions in the objective function (i.e. the number of design goals or decision criteria) grows. While orthogonal axis plots may be used to represent the results from problems with two-dimensional (i.e. the Pareto optimal set is a line boundary) and three-dimen-



(a)



(b)



(c)

**Figure 3.28** The Aerospace Corporation TechSat 21 Baseline Design Local Trade Space Pareto Boundary Approximation by 1-DOF (a), 2-DOF (b), and 3-DOF (c) Variants of the Multiobjective Multiple Solution Simulated Annealing Algorithm

sional (i.e. the Pareto optimal set is a surface boundary) objective functions, such plots may not be used for objective functions with four or more dimensions (i.e. Pareto optimal set is a hyper-surface boundary). In these cases, "radar plots" can be used to graphically compare two or more Pareto optimal architectures.

Table 3.19 lists the objective function metric values for five different Pareto optimal TechSat 21 space-based radar system architectures when the design goals are to simultaneously:

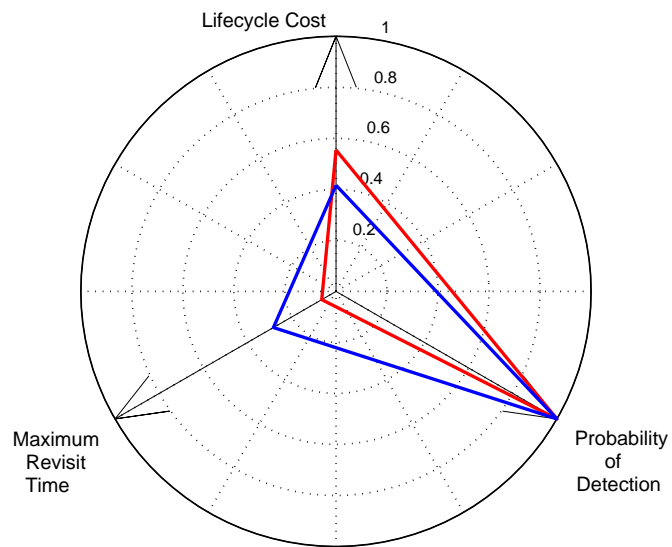
1. Minimize Lifecycle Cost,
2. Maximize Target Probability of Detection, and
3. Minimize Theater Revisit Time.

**TABLE 3.19** Five Sample Pareto Optimal TechSat 21 System Architectures

Objective Function Metric	Pareto Optimal Architecture 1 (Red Line)	Pareto Optimal Architecture 2 (Blue Line)	Pareto Optimal Architecture 3 (Green Line)	Pareto Optimal Architecture 4 (Cyan Line)	Pareto Optimal Architecture 5 (Magenta Line)
Lifecycle Cost (\$B)	12.0	9.02	4.54	4.80	4.00
Target P(d)	0.999	0.999	0.999	0.790	0.600
Max. Revisit Time (min.)	2.0	8.8	9.9	2.0	8.9

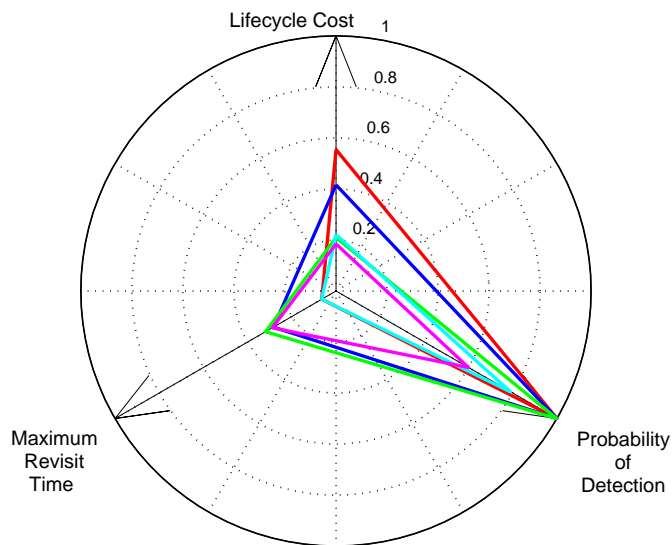
The first two architectures are compared graphically in the three-dimensional radar plot in Figure 3.29a, and all five architectures are compared graphically in Figure 3.29b. The metric values on each axis of the radar plot are normalized with respect to their maximum observed value within the trade space. As one can see, the first architecture (red line) exhibits a shorter theater revisit time than the second architecture (blue line), but at the expense of having a higher lifecycle cost. Both architectures provide the same target probability of detection. This same type of radar plot may be used to visually compare two or more Pareto optimal architectures for an optimization problem with an objective function containing any number of dimensions. The number of axes in the radar plot will correspond to the number of dimensions in the objective function.





(a)

Comparison of 5 Pareto Optimal TechSat 21 Designs



(b)

**Figure 3.29** Radar Plots Comparing the First Two (a) and All Five (b) Sample Pareto Optimal TechSat 21 Architectures from Table 3.19

### **Multiobjective Optimization Summary**

In Section 3.5.2, several new techniques have been developed as the MMDOSA methodology for the conceptual design of distributed satellite systems continues to be introduced. First, the problem scope was greatly increased by expanding from single objective optimization problems to multiobjective optimization problems. This allowed more accurate modeling and capturing of the trades involved in real-world DSS conceptual design problems. Next, both an algorithm that searches for a single Pareto optimal architecture and an algorithm that searches for multiple Pareto optimal architectures during a single run were developed. Lastly, the utility in knowing the true global Pareto boundary of a DSS design trade space was discussed and several methods were explored for approximating this global Pareto boundary with only a limited knowledge of the full DSS trade space. The Pareto boundary empowers the systems engineer to capture trades between any desired design parameters along the most efficient set of conceptual design solutions for a DSS. With the conclusion of Section 3.5, both single objective and multiple objective algorithms are now available for the MMDOSA methodology to use in the conceptual design of distributed satellite systems.

### **3.6 MMDOSA Step 5 – Interpret Results (Sensitivity Analysis)**

Once the single objective and multiobjective MDO algorithms developed and introduced in Section 3.5 have been applied to the DSS conceptual design problem, the results must be analyzed and interpreted to ensure that the output makes sense and that the GINA model is of sufficient fidelity to capture the relevant trades within the conceptual design problem. This process began in step two of the MMDOSA methodology with the application of univariate studies to investigate how a single system architecture attribute (i.e. simulation output variable or GINA metric) varies as a function of a single input parameter (i.e. design vector or constants vector variable). After the MDO algorithms have been applied in step four of the MMDOSA methodology, an analysis of variance may be executed on the examined system architectures to assess the relative impact of the different

design decisions the systems engineer has control over (i.e. the design vector and constants vector variables) on the design decision criteria (i.e. the system metrics of interest). In this manner, analysis of variance represents a sensitivity analysis tool for the interpretation of the MDO results.

### 3.6.1 Analysis of Variance (ANOVA)

A common problem in the field of statistics concerns how to detect differences in the average performance of groups of items tested. A tool commonly used to solve this problem is a technique named analysis of variance (ANOVA). ANOVA works by separating the total variation displayed by a set of observations measured by summing the squares of the deviations from the mean into components associated with defined sources of variation, such as the elements of a design vector [Logothetis and Wynn, 1989]. During the conceptual design of a DSS, the systems engineer would like to know which elements of the design vector have a significant effect on the metrics of interest and which design vector elements dominate the design. ANOVA provides the systems engineer with a tool to answer these questions.

This section illustrates the steps involved in a typical ANOVA procedure. The variables used in this section are defined in Table 3.20.

**TABLE 3.20** ANOVA Terminology

<b>Variable</b>	<b>Definition</b>
$SS$	Sum of the squares
$T$	Sum of all measurement observed outputs
$N$	Total number of observations
$\Gamma$	Design Vector
$\gamma$	Design Vector Variable
$n_{\gamma_i}$	Number of observations where variable $\gamma$ is a setting $i$
$RI$	Relative Influence

Recall that the design vector for a DSS is defined as:

$$\Gamma = [\gamma_1 \ \gamma_2 \ \gamma_3 \ \dots \ \gamma_n] \quad (3.50)$$

where each  $\gamma$  represents a different independent variable in the conceptual design problem. ANOVA will be used to look for variations in the output metrics due to different variables within the design vector.

The first step in the ANOVA process involves the sum of the squares computations. Sum of the squares computations measure the variation of a subset of observations about the arithmetic mean of the entire set of observations. For the DSS conceptual design problem, this is the variation of the average of observations under each design variable setting around the average of all observations. The total sum squares  $SS_T$  is the sum of the squared differences between each observation  $y_i$  (i.e. a system metric such as CPF) and the arithmetic mean  $\bar{T}$  of the entire set of observations:

$$SS_T = \sum_{i=1}^N (y_i - \bar{T})^2 = \left[ \sum_{i=1}^N y_i^2 \right] - \frac{T^2}{N} \quad (3.51)$$

The sum of the squares  $SS_\gamma$  for each variable in the design vector is:

$$SS_\gamma = \sum_{i=1}^n n_{\gamma_i} (\bar{y}_{\gamma_i} - \bar{T})^2 \quad (3.52)$$

where  $n$  is the total number of possible settings for design variable  $\gamma$  (ex. orbit = 1AU-5AU, integer),  $n_{\gamma_i}$  is the number of observations for which design variable  $\gamma$  is at setting  $i$  (ex. orbit=2AU), and  $\bar{y}_{\gamma_i}$  is the arithmetic mean of all the observations for which design variable  $\gamma$  is at setting  $i$ .

With this information, the percentage of the total variation observed that can be attributed to each design vector variable may now be computed. This relative percentage contribution or relative influence  $RI$  indicates the relative power of each design variable to reduce

variation. The *RI* may be computed percentage wise for each element of the design vector via Eqn. 3.53:

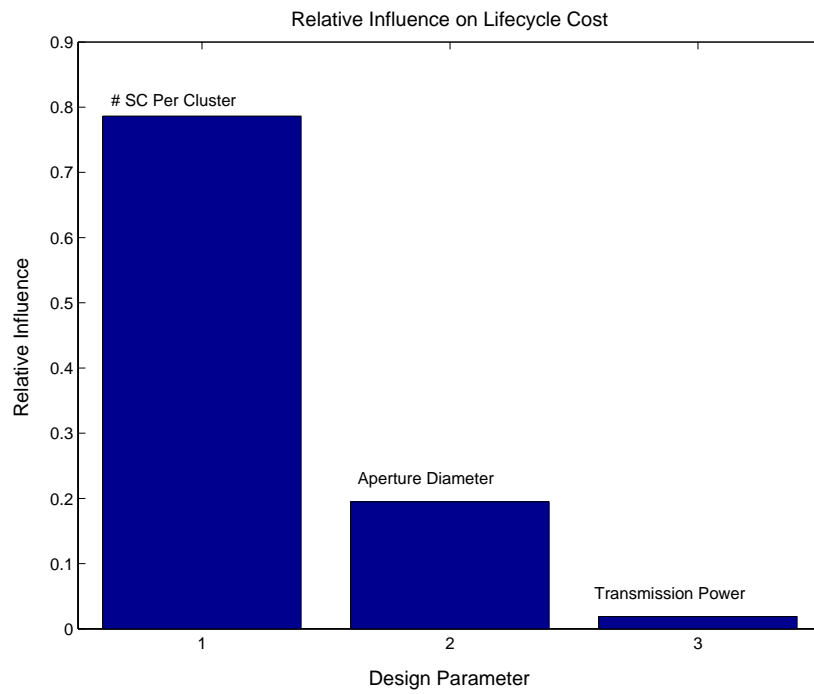
$$RI = \left[ \frac{SS_y}{SS_T} \right] \times 100\% \quad (3.53)$$

### 3.6.2 ANOVA Applied to the TechSat 21 Case Study

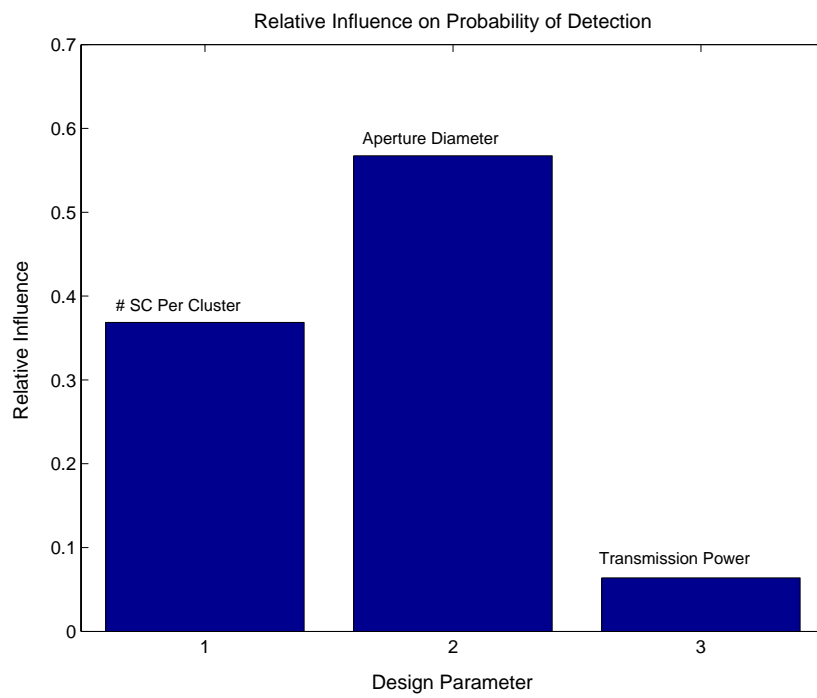
This section briefly examines a sample application of ANOVA to the TechSat 21 case study presented in Chapter 5 to illustrate its utility within the MMDOSA methodology. Interpreting the results from the MDO algorithms in step four of the MMDOSA methodology requires a sensitivity analysis of the design parameters of interest on the system attributes. This is done via the ANOVA method. Unlike a local sensitivity analysis – which is valid only in the immediate vicinity of a single point design, requires finite differencing, and is thus computationally expensive and time consuming – ANOVA provides the systems engineer with quantitative information on the relative influence of each design variable on the system attribute or metric of interest using the information already gathered from the execution of the MDO algorithm.

Figure 3.30 illustrates the ANOVA results for the Aerospace Corporation CDC TechSat 21 point design. The number of satellites per cluster exerts by far the greatest relative influence on the system lifecycle cost, while the radar antenna aperture diameter exerts the greatest relative influence on the target probability of detection. Notice that relative to these two design vector variables, the radar antenna transmission power exerts little influence on the system attributes of lifecycle cost and performance.

ANOVA provides the systems engineer with information on which design parameters give the most leverage over various attributes of the system. The ANOVA results may also be used to guide technology investment decisions. For example, if the TechSat 21 program has a limited technology investment budget and reaching the desired target probability of detection is the foremost priority, then it appears from the ANOVA results that those funds



(a)



(b)

**Figure 3.30** ANOVA Results for The Aerospace Corporation CDC Trade Space with Respect to Lifecycle Cost (a) and Target Probability of Detection (b)

would be best spent on technologies which will yield light weight and low cost large diameter radar transmission antennas.

### **3.6.3 ANOVA Summary**

ANOVA provides a global sensitivity analysis tool in step five of the MMDOSA methodology to aid in the interpretation of the MDO algorithm results from step four of MMDOSA. First, ANOVA tells the systems engineer which elements of the design vector exert the greatest relative influence on the metrics of interest for a particular DSS. For example, it was found that the aperture diameter exerts by far the greatest relative influence on the target probability of detection metric for TechSat 21. A good systems engineer can use this information effectively during the conceptual design phase of a program to dramatically improve both the performance and economy of a mission system architecture.

The results for ANOVA may also be used for technology roadmapping. A common question encountered by both systems engineers and program managers is "Given a limited number of resources (i.e. budget), in which design vector elements (i.e. technologies) should I invest in now (i.e. at the beginning of the program) to exert the greatest downward pressure on the cost per function metric?" For the TechSat 21 case study example, the ANOVA results point towards the number of spacecraft per cluster as being the most important design variable in terms of minimizing lifecycle cost. In this case, the highest priority of any technology development budget aimed at reducing lifecycle cost and initiated at the beginning of the program should be investing in key technologies which will enable adequate target probability of detection with as few satellites per cluster as possible. In this manner, ANOVA can provide a powerful tool for developing a technology roadmapping strategy for distributed satellite systems.

### 3.7 MMDOSA Step 6 – Iterate

After steps four and five of the MMDOSA methodology, the systems engineer may have enough information to identify a family(s) of system architectures to carry forward into the next phase of the design process. Before doing so, however, the systems engineer should carefully review the results from the MDO algorithms and the ANOVA sensitivity analysis to determine if it may be possible to find even better system architectures. This involves an iterative process of examining the results from MMDOSA steps four and five, making the appropriate changes to the GINA model and/or the MDO algorithms, and re-running the optimization(s).

The systems engineer can do many different things to improve the performance of the MDO search as a function of the structure of the DSS conceptual design problem, the nature of the GINA model, and the size of the trade space. For example, the sample ANOVA sensitivity analysis on the Aerospace Corporation CDC design trade space in Section 3.6.2 indicated that the aperture diameter exerted the greatest relative influence on the target probability of detection metric. This suggests that because aperture diameter appears to be the driving parameter for system performance, the fidelity of the radar performance module within the GINA model should be verified – especially the mathematical relationships between the aperture diameter of the radar antenna and the target probability of detection. In this manner, the ANOVA results may guide the systems engineer to identify which modules within the GINA model are most important – which modules should be reexamined and perhaps improved.

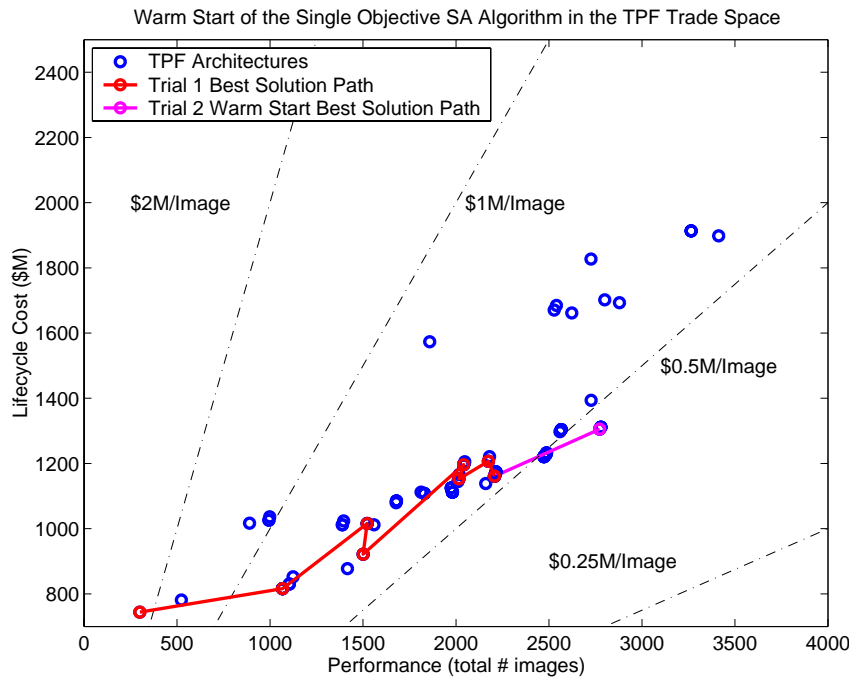
In addition to improving model fidelity, the systems engineer may modify certain parameters within the single objective and multiobjective simulated annealing algorithms to improve the performance of these algorithms. For example, if the simulated annealing algorithm appears to be clustering within a single portion of the trade space, the cooling schedule may be altered to confirm that this region of the trade space is indeed a global optimum rather than a local optima. Additionally, if the design vector is very large, the



DOF parameter (Section 3.5.1), which determines how the simulated annealing algorithm moves from one neighboring architecture to the next, may be varied to see if better system architectures can be found. Figure 3.20 in Section 3.5.1 illustrates how the performance of the single objective simulated annealing algorithm may vary as a function of the DOF parameter.

Another iterative strategy that can be employed to improve algorithm performance is to use the current best solution (i.e. final system architecture) from a previous run of an MDO algorithm as the starting point (i.e. initial system architecture) for another run of the same MDO algorithm. This is known as "warm starting." In the case of a single objective optimization problem, the warm start architecture will probably be closer to the global optimum than the randomly selected initial starting point system architecture, and thus the second run of the same single objective MDO algorithm from the new starting point is more likely to find the true global optimum. In the case of a multiobjective optimization problem, the warm start architecture will probably be closer to the Pareto boundary than the randomly selected initial starting point system architecture, and thus the second run of the same multiobjective MDO algorithm from the new starting point is likely to find more Pareto optimal system architectures.

Figure 3.31 illustrates an example of warm starting the single objective (i.e. minimize the cost per image metric) simulated annealing algorithm within the TPF trade space. Table 3.21 lists the architecture data for this example. The first run of the algorithm begins at an architecture with a very poor cost per image of nearly \$2.5M and converges on an architecture with a cost per image of \$526K. By warm starting the algorithm from this \$526K/image architecture, the \$500K/image threshold is passed and a TPF system architecture with the very low cost per image of \$471K is found. As one can see, warm starting the simulated annealing algorithm from the best solution of a previous trial can improve the quality of the final DSS architectures converged upon.

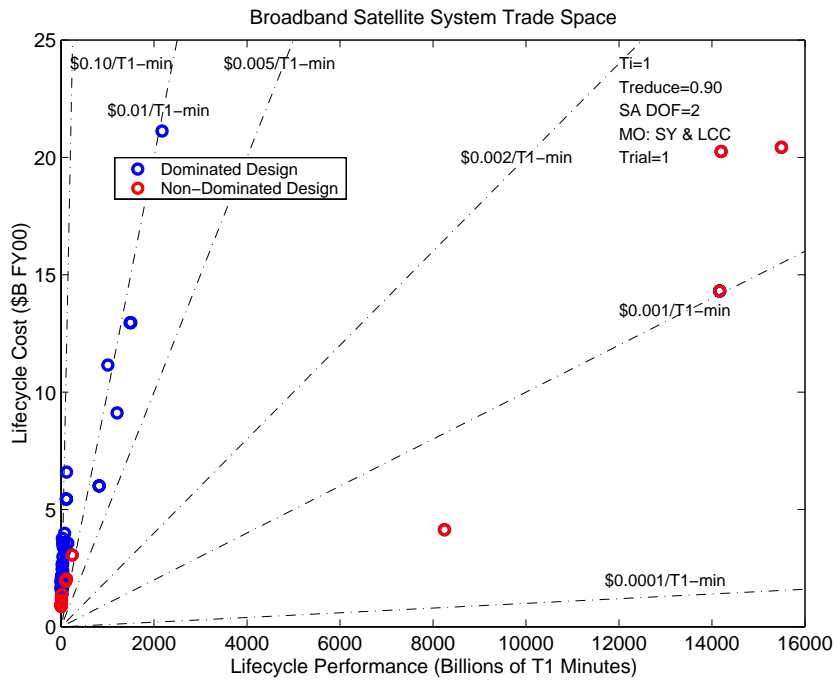


**Figure 3.31** Warm Start of the Single Objective Simulated Annealing Algorithm in the TPF Trade Space

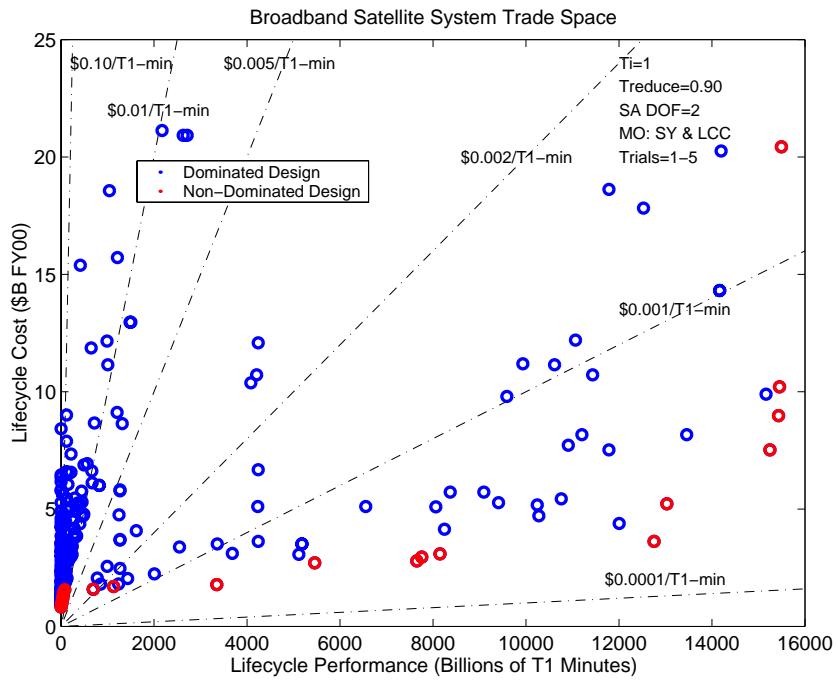
**TABLE 3.21** Data From the TPF Warm Starting Example in Figure 3.31

State	Orbit (AU)	Number Apertures	Collector Connectivity/ Geometry	Aperture Diameter (m)	Cost Per Image (\$K)
Trial 1 Initial Architecture	1.0	4	SCI-1D	1	2475
Trial 1 Final Architecture	4.5	8	SCI-2D	3	526
Trial 2 Warm Start Initial Architecture	4.5	8	SCI-2D	3	526
Trial 2 Warm Start Final Architecture	4.5	8	SCI-2D	4	471

Finally, the systems engineer may simply run more trials of the same MDO algorithms from different random starting points to collect more data. This will increase the probability of finding the best system architectures in a single objective optimization and of finding the true Pareto optimal set of architectures in a multiobjective optimization. Figure 3.32 illustrates a two-dimensional Pareto optimal set of system architectures for the broadband satellite communications case study presented in Chapter 6. In this case, the systems engineer wishes to simultaneously maximize system performance and minimize



(a)



(b)

**Figure 3.32** Broadband 2-Dimensional Pareto Optimal Set of System Architectures After One Trial (a) and Five Trials (b) of the Multiobjective Multiple Solution Simulated Annealing Algorithm

lifecycle cost. Figure 3.32a illustrates the Pareto front found after one trial of the multiobjective multiple solution simulated annealing algorithm, and Figure 3.32b illustrates the Pareto front from five cumulative trials of the same algorithm. As one can see, running multiple trials of the same MDO algorithm increases both the magnitude and fidelity of the Pareto optimal set of DSS architectures.

To summarize, step six in the MMDOSA methodology involves an iterative process of examining the results from MMDOSA steps four and five, making the appropriate changes to the GINA model and/or the MDO algorithms, and re-running the optimization(s). Possible modifications include increasing the fidelity of the critical models as identified by the ANOVA sensitivity analysis, changing the cooling schedule and/or DOF parameter within the simulated annealing algorithm, warm starting a new optimization run from the final answer of a prior optimization run, and running additional trials of the same optimization algorithm. By taking an iterative approach to searching the trade space, the systems engineer gains confidence that the architectures converged upon by the MDO algorithms are indeed among the best system architectures within the full DSS trade space. In this manner, step six of the MMDOSA methodology leads to step seven.

### **3.8 MMDOSA Step 7 – Converge on Best System Architectures**

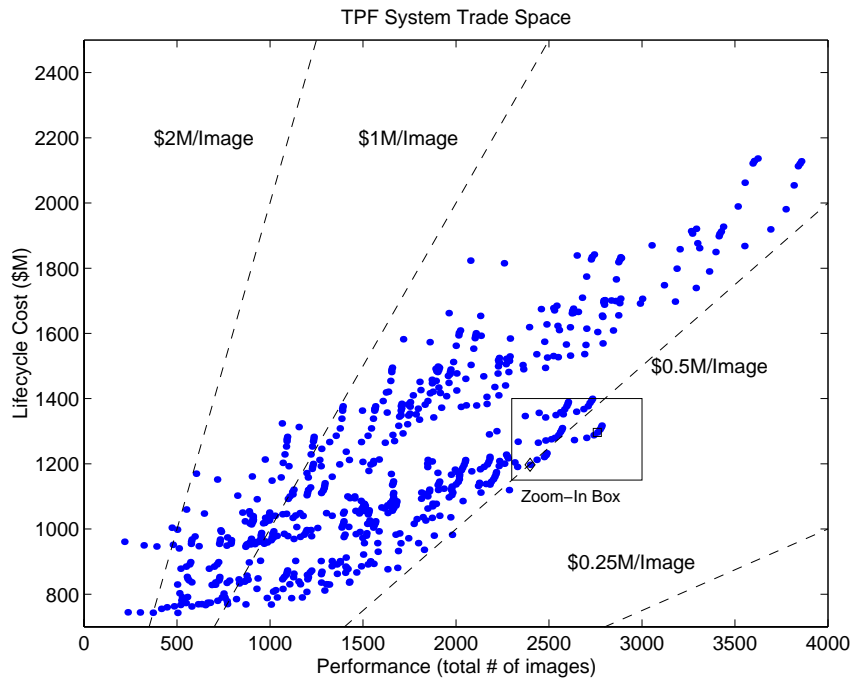
In the final step of the MMDOSA methodology, system architectures to be carried forward into the next phase of the space systems design and development process are identified based upon the iterative optimization and sensitivity analysis results in MMDOSA steps four through six. In the case of a single objective DSS conceptual design problem, the recommended architectures are the best family(s) of architectures with respect to the metric of interest found by the single objective simulated annealing algorithm. In the case of a multiobjective DSS conceptual design problem, the recommended architectures are those that comprise the Pareto optimal set with respect to the selected decision criteria as found by the multiobjective multiple solution simulated annealing algorithm.

If the customer wishes to optimize the DSS with respect to a single metric, then the architectures the systems engineer should carry forward into the next phase of the space systems design and development process are the architectures which comprise the best family(s) of conceptual design solutions. Recall from Section 1.1 and Section 2.3.2 that the number of best system architectures within a DSS trade space varies as a function of the size of the trade space. Within the context of exploring the single objective trade space of distributed satellite systems, the number of good solutions within the trade space will be defined as consisting of the top 0.1%-10% of design solutions in the trade space, depending on the size of the trade space, as defined by the metric of interest to the customer. The number of best solutions will be defined as consisting of the top 0.01%-1% of design solutions in the trade space. Table 3.22 lists the number of good and best system architectures within the trade space for each of the three DSS conceptual design case studies in Chapters 4-6. Notice how the percentage of good and best solutions must scale inversely with the magnitude of the trade space in order to keep the total number of best design architectures down to a manageable size for a real-world space system design program.

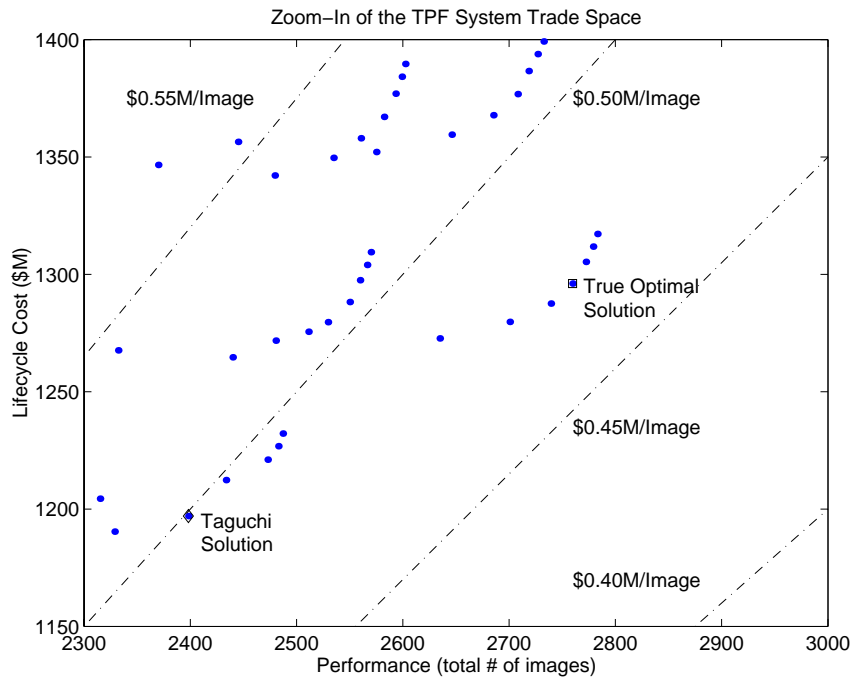
**TABLE 3.22** Number of Good and Best Solutions in the Three Case Studies

Case Study	Trade Space Magnitude	Good Architectures	Best Architectures
Terrestrial Planet Finder	640	Top 10% 64	Top 1% 6
TechSat 21	732,160	Top 0.1% 732	Top 0.01% 73
Broadband	42,400	Top 1% 424	Top 0.1% 42

According to the definition in Table 3.22, the TPF mission trade space contains approximately six best architectures that should be carried forward into the next phase of the design process. Figure 3.33a illustrates the complete TPF global trade space for the design vector defined in Table 3.11, and Figure 3.33b zooms in on the optimal design region according to the cost per image metric. Notice that several arcs, or families, of system architectures exist within this region. The optimal family of TPF system architectures



(a)



(b)

Figure 3.33 TPF Global Trade Space (a) and Zoom-In on the CPI Optimal Design Region (b)

is the right-most arc, which is further zoomed in on in Figure 3.34. If this TPF case study was being carried out to the next stage, all of the design architectures in the local family of solutions composing the arc in which the true optimal solution lies merit further detailed analysis as the difference between their CPI's is within the uncertainty of the computations. The performance of all these system architectures lies within 150 images of each other, while total lifecycle cost varies by only \$45M. Notice that three of the four design vector elements remain constant through the arc – collector connectivity/geometry (i.e. structurally connected, two dimensional), number of apertures (i.e. 8), and aperture diameter (i.e. 4 m). The only design vector parameter that varies through the arc is the heliocentric orbital radius, with the true optimal solution occurring at 4 AU. Based on these results, the family of solutions defined by this arc merits further detailed study in the next mission design phase. This example illustrates how the application of single objective MDO can help focus the design effort during the conceptual design phase of a program.

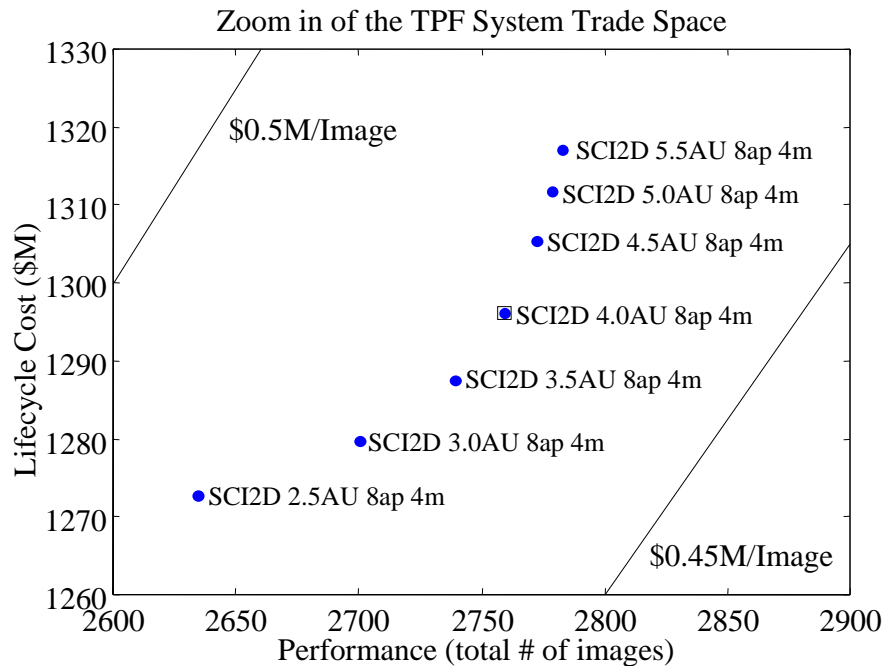
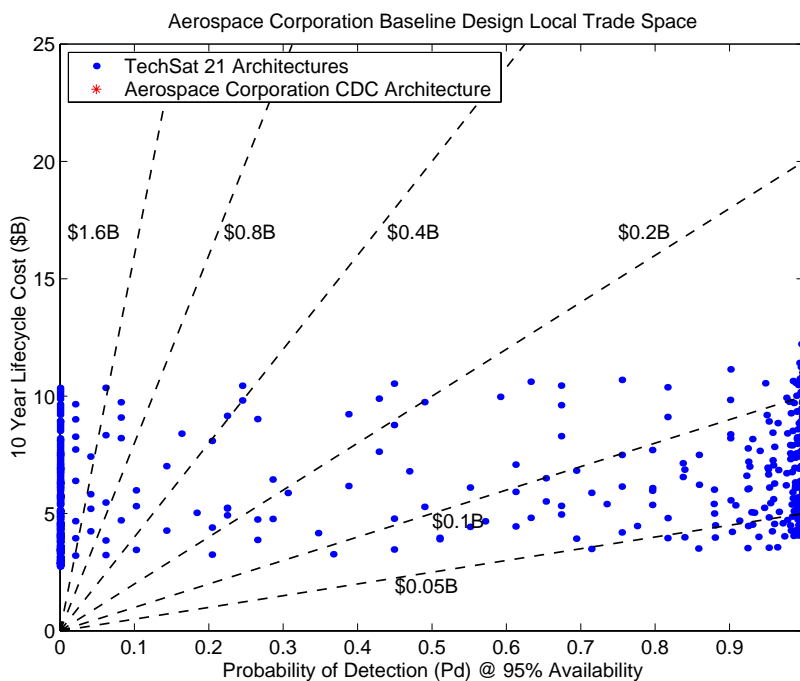


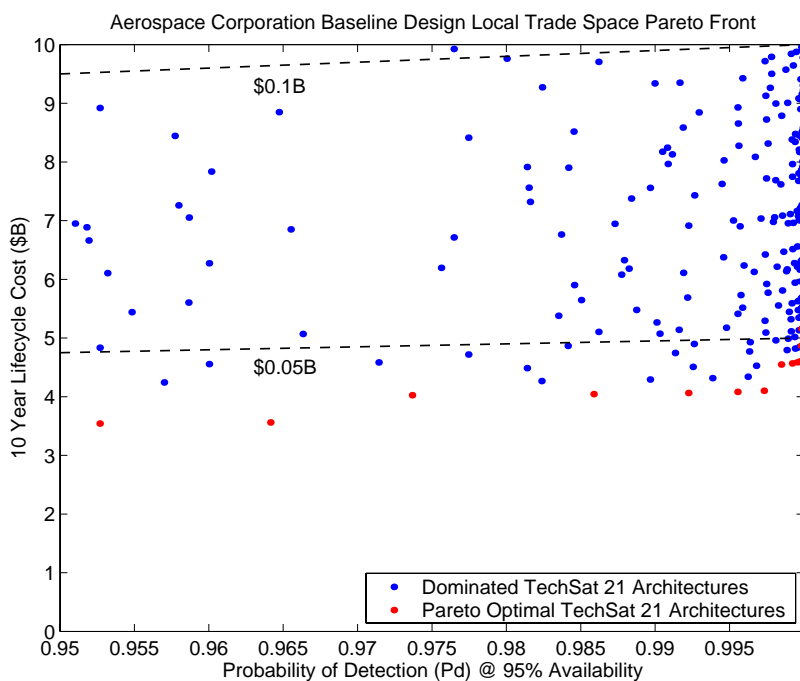
Figure 3.34 TPF Optimal (i.e. Best) Family of System Architectures

If the customer wishes to optimize the DSS with respect to multiple metrics, then the architectures the systems engineer should carry forward into the next phase of the space systems design and development process are the architectures which comprise the Pareto optimal set. Figure 3.35a illustrates the trade space for the TechSat 21 Aerospace Corporation CDC design problem, and Figure 3.35b illustrates the Pareto optimal system architectures located within the high performance region of the trade space. Table 3.23 lists some of the key parameters for the architectures within the Pareto optimal set. Recall that the Pareto optimal set represents a set of system architectures in which the systems engineer cannot improve one metric without harming at least one other metric. In the two-dimensional TechSat 21 trade space (i.e. maximize system performance and minimize lifecycle cost) of Figure 3.35, this means that within the Pareto optimal set the systems engineer cannot improve the performance of the space-based radar system without also increasing lifecycle cost. Thus, in a two-dimensional trade space, the Pareto optimal set represents the most efficient boundary that captures the trades between the DSS conceptual design decision criteria. In this manner, the Pareto optimal set contains the best system architectures to carry forward into the next phase of the space systems design and development process. If the Pareto optimal set is large and contains too many architectures to carry forward, then different methods may be used to identify which architectures within the Pareto optimal set should remain under consideration. One such method is budget capping. In budget capping, all of the architectures in the Pareto optimal set whose lifecycle cost appears to exceed the anticipated program budget are eliminated from consideration. For example, if the TechSat 21 budget was capped at \$4B over the life of the mission, then system architectures three through nine in Table 3.23 would be eliminated from consideration and only architectures one and two would be carried forward into the next design phase. Other methods for identifying the best system architectures within a large Pareto optimal set include multiattribute utility theory (MAUT) [de Neufville, 1990], uncertainty analysis differentiation [Walton & Hastings, 2001], architecture flexibility [Saleh & Hastings, 2000], and architecture policy implications [Weigel & Hastings, 2001].





(a)



(b)

**Figure 3.35** The TechSat 21 Aerospace Corporation CDC Design trade Space (a) and the Pareto Optimal Architectures Located Within the High Performance Region of the Trade Space (b)

**TABLE 3.23** Pareto Optimal TechSat 21 Architectures from Figure 3.35b

Architecture	1	2	3	4	5	6	7	8	9
# Satellites Per Cluster	4	4	4	4	4	4	4	4	4
Aperture Diameter	2.0	2.0	2.5	2.5	2.5	2.5	2.5	3.0	3.0
Radar Transmission Power	900	1000	600	700	800	900	1000	700	800
Target Probability of Detection	0.953	0.964	0.974	0.986	0.992	0.996	0.997	0.998	0.999
Lifecycle Cost (\$B)	3.54	3.56	4.02	4.04	4.06	4.08	4.10	4.55	4.57
Max. Theater Revisit Time (min.)	15	15	15	15	15	15	15	15	15

To summarize, system architectures to be carried forward into the next phase of the space systems design and development process are identified in step seven of the MMDOSA methodology. In single objective DSS conceptual design problems, the recommended architectures are the best family(s) of architectures with respect to the metric of interest. In multiobjective DSS conceptual design problems, the recommended architectures are those that comprise the Pareto optimal set with respect to the selected decision criteria. This represents the final step of the MMDOSA methodology. At this point, the DSS mission is ready to move forward into Phase B of the space systems design and development process – preliminary design of the best subset of architectures within the global trade space. Phase A of the design process – conceptual design, for which the MMDOSA methodology has been created – is now over.

### 3.9 MMDOSA Summary – Tying it All Together

MMDOSA – the multiobjective, multidisciplinary design optimization systems Architecting methodology for the conceptual design of distributed satellite systems – provides a framework for mathematically modeling distributed satellite systems as mathematical optimization problems to enable an efficient search for the best, as defined by the metric(s) of interest to the customer, system architectures within the DSS conceptual design trade space. This chapter has detailed the MMDOSA methodology step by step, including the origin of each step, the mathematics behind each step, and the utility of each step to the systems engineer developing a conceptual design for a DSS. Figure 3.1 on page 76 illustrates the flow of the seven steps in the MMDOSA methodology, and how these steps fit

together to provide a systems engineering tool for the conceptual design of distributed satellite systems.

In step one, the GINA methodology is applied to create a simulation model of the DSS. By modeling the space system as an information network, quantitative trades directly relating lifecycle performance to lifecycle cost may be performed. In step two, univariate studies are performed in which one parameter in a baseline design architecture is varied along a range of values to see how the attributes of the system change. These univariate studies provide the systems engineer with an initial feel for the DSS trade space and also provide a means of identifying model fidelity problems. In step three, a random sample of architectures within the DSS trade space are analyzed. This random sample provides the systems engineer with a glimpse of the global trade space, identifies initial bounds on each of the metrics the design will be optimized with respect to, and yields statistical data that will be used to tailor the MDO algorithms. In step four, single objective and/or multiobjective MDO algorithms are exercised to search for the best system architectures within the global DSS trade space. The single objective simulated annealing algorithm identifies the best family(s) of system architectures on the basis of the metric of interest, while the multiobjective simulated annealing algorithms identify the Pareto optimal set of system architectures with respect to all of the decision criteria. In step five, an ANOVA sensitivity analysis is performed to identify the most influential design variables and the models which need the highest fidelity. Step six involves an iterative process of examining the results from MMDOSA steps four and five, making the appropriate changes to the GINA model and/or MDO algorithms, and re-running the optimization(s). By taking an iterative approach to searching the trade space, the systems engineer gains confidence that the system architectures converged upon by the MDO algorithms are indeed among the best system architectures within the full DSS trade space. Finally, in step seven the system architectures to be carried forward into the next phase of the design process are identified. At this point, the DSS mission is ready to move forward into Phase B of the space systems design and development process – preliminary design of the best subset of architectures within the global trade space. With the conclusion of Chapter 3, the origin, mathematics,

and mechanics of the MMDOSA methodology have been fully developed and presented. The next three chapters present case studies illustrating the application of MMDOSA to the conceptual design of civil, military, and commercial distributed satellite systems.

# Chapter 4

## TERRESTRIAL PLANET FINDER MISSION CASE STUDY AND RESULTS

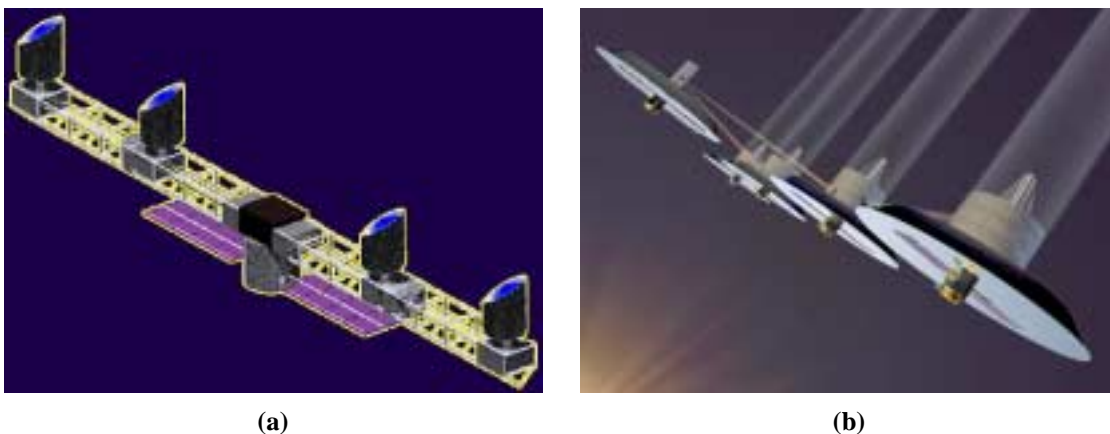
*A great deal of effort will be required over the next few years to arrive at a design that optimizes the performance of the entire system while minimizing cost and complexity. – The TPF Science Working Group*

Chapter 4 presents the first class of space missions MMDOSA may be applied to – civil missions. Both single objective and multiobjective MDO algorithms are applied to NASA's Terrestrial Planet Finder (TPF) mission to search for the most cost-effective architectures within the trade space. This chapter demonstrates step-by-step the application of the MMDOSA methodology to the conceptual design of the TPF mission, and concludes by illustrating how MMDOSA found better system architectures during the conceptual design phase than the existing point design architectures for this mission.

### **4.1 Introduction to the Terrestrial Planet Finder Mission**

TPF is one in a series of missions as part of NASA's Origins Program, whose goal is to answer fundamental questions regarding the origin of life in the universe. Specifically, TPF will be the first spacecraft to directly detect the existence of Earth-like planets around neighboring stars [Beichman, 1998]. Using the principles of nulling interferometry, TPF will suppress light from the parent star by a factor of one million while maintaining good transmissivity in the parent star's habitable zone between 0.5 AU and 3 AU. Once identi-

fied, TPF will carry out spectroscopic observations of these extra-solar planets to search for chemical compounds capable of supporting life [Beichman et al, 1999]. The TPF program is currently in the conceptual design phase, and several widely varying architectures from structurally connected (Figure 4.1a) to separated spacecraft (Figure 4.1b) interferometers are being considered. The remainder of this chapter details how MMDOSA has been applied to the conceptual design of the TPF mission.



**Figure 4.1** A Structurally Connected (a) and Separated Spacecraft (b) TPF Architecture [Beichman et al, 1999]

## 4.2 MMDOSA Step 1 – Create the GINA Model

GINA – the Generalized Information Network Analysis methodology – is a systems engineering and architecting framework for the comparative analysis of distributed satellite systems [Shaw et al, 2001]. GINA enables the comparison of different system architectures for a given mission. The foundation behind GINA is the assertion that most satellite systems are information disseminators that can be represented as information transfer networks [Shaw et al, 2000]. A summary of the procedural steps in the GINA methodology is presented in Section 3.2. Through these steps, GINA allows the systems engineer to make quantitative trades at the conceptual design level by directly relating lifecycle performance to lifecycle cost. The remainder of this section details the application of the GINA methodology to the TPF mission.

### 4.2.1 GINA Step 1 – Definition of the Mission and CDP Objectives

In defining the mission objective, the systems engineer must explicitly state what the real application of the system is in terms of the user needs. To do this, the customer of the system must first be identified. In the case of the TPF mission, there exist two separate customers. The first is the space science and astronomy community. This community will receive all of the scientific data, and would ideally prefer a system that generates the greatest amount of high quality data from as many targets as possible. Because such a system will most likely be financed by a single federal government or through a combination of governments, the second customer is the taxpayer community. In contrast to the scientists, this community would prefer the least expensive system that meets a set of minimum performance requirements. The conflict between these two opposing communities will be captured and illustrated in the GINA model.

Once the customer has been identified, the mission objective may be defined. The official NASA Origins program mission objective for TPF is:

To study all aspects of planets, ranging from their formation and development in disks of dust and gas around newly forming stars to the presence and features of those planets orbiting the nearest stars. Specifically, to conduct a search for Earth-like planets in star systems located within 15 parsecs of our solar system. – The TPF Science Working Group [Beichman et al, 1999]

Close inspection of this statement reveals what the system must accomplish to meet the needs of the user, in this case space scientists. First, the system must detect Earth-like planets in the habitable zone around nearby stars. This means the system must possess the capability to detect a faint signal source (i.e. the planet) nearby a large noise source (i.e. the parent star). For TPF, this will be done using a Bracewell nulling interferometer [Beichman et al, 1999]. Second, in order to characterize the planets that are discovered to determine whether they contain the elements conducive to life, spectroscopy will be required to measure the chemical composition of the planets' atmosphere. Thus, defining the mission objective in terms of the user needs begins to elucidate the form and function of the space system to be designed. It also provides the top-level requirements of the sys-

tem. In the case of TPF, this equates to being able to isolate a planet from its parent star, which may be located up to 15 parsecs from Earth, and being able to actually measure the chemical composition of the planet's atmosphere.

Next, the systems engineer must explicitly define the objective to be achieved during the conceptual design phase to remove all ambiguity concerning what will have been accomplished by the end of this phase of the design process. For the TPF case study, the CDP objective was:

To develop a methodology for the comparison of TPF system architectures spanning from structurally connected to separated spacecraft interferometers. [Miller et al, 2000]

It is important to stress that the conceptual design phase goal is *not* to create a single point design, but rather to create a tool that may be used to model, analyze, and compare different system architectures for the NASA Origins TPF mission.

#### **4.2.2 GINA Step 2 – Information Network Representation**

Modeling the TPF system as an information network entails defining all of the origin-destination (O-D) pairs in the system (Figure 4.2). For TPF, the origin is the target star system(s) and the destination is where the data is assembled; either in a computer on the spacecraft itself, at a groundstation, or at a control/data center.

In the GINA methodology, the capability of an architecture is characterized by four quality of service parameters that relate to the detection process, and to the quantity, quality, and availability of the information that is processed through the network. These four parameters are signal isolation, information rate, information integrity, and the availability of these services over time [Shaw et al, 2000]. Once formulated, these four parameters serve as the minimum instantaneous capability requirements the system must meet to satisfy the customer. These capability metrics assess how well the customer requirements are met at an instantaneous point in time.

- Isolation



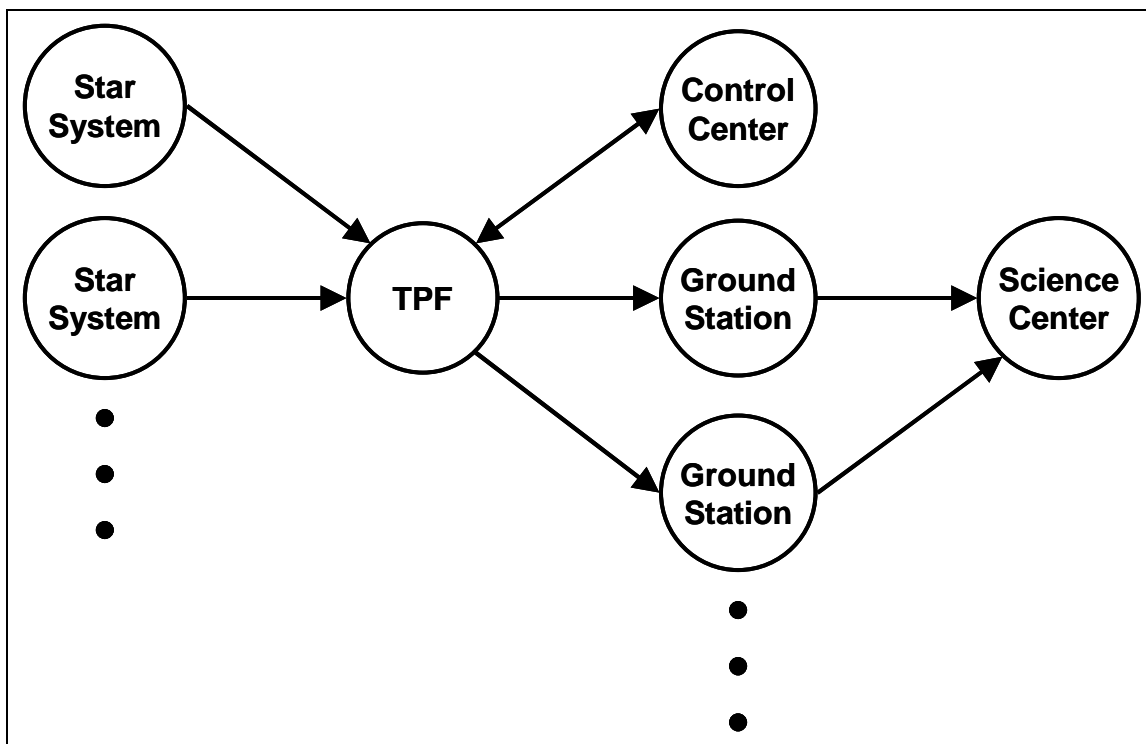


Figure 4.2 The TPF System Modeled as a Network

Isolation refers to the ability of a system to isolate and distinguish information signals from different signal and noise sources within the network. For TPF, the system’s angular resolution, which is a function of the maximum vector baseline between a pair of collectors, determines the smallest sized objects the interferometer can image and discriminate between in the field of view. Additionally, the null depth and shape of the transmissivity function quantify the ability of the system to null out the parent star and see planets in the habitable zone.

- Integrity

Integrity is a measure of the quality of the information being transferred through the network, typically quantified as the bit error rate. In the case of TPF, the integrity of an individual image is a function of the signal-to-noise ratio (SNR) required to obtain that image. The integrity requirements may vary with each mode of operation of TPF as illustrated in Table 4.1.

- Rate

Rate measures the speed at which the system transfers information between the nodes (i.e. sources and sinks or O-D pairs) in the network, and is typically measured in bits per second for a network. For TPF, the rate is the total number of images the system can produce per unit time and varies for each of the three modes of operation – surveying, imaging, and spectroscopy.

- Availability

Availability characterizes the instantaneous probability that information is being transferred through the network between all of the sources and sinks. TPF targets close to the sun whose imaging needs violate sun avoidance angles within the optical train reduce the availability of the system. The actual imaging time versus the time to complete other tasks such as calibration and retargeting also affects the availability of the system.

At this point, the systems engineer must decide which of these capability parameters will become hard requirements the system must meet to be considered feasible and which parameters will serve to distinguish between architectures. For TPF, the isolation, integrity, and availability parameters are hard requirements every potential architecture must meet, while the rate will differ between architectures. Table 4.1 summarizes the capability parameters as applied to TPF along with the requirements placed on these parameters by the NASA Jet Propulsion Laboratory.

**TABLE 4.1** The GINA Capability Quality of Service Metrics for the TPF GMTI Mission

Capability	Equivalent Design Parameter	NASA Requirement
Isolation	Angular Resolution	2.5 - 20 milli-arcsec
	Null Depth	$10^{-6}$
Integrity	Signal-to-Noise Ratio	
	– Survey Mode	5
	– Medium Spectroscopy Mode	10
	– Deep Spectroscopy Mode	25
Rate	Images Per Unit Time	Free Variable (Maximize)
Availability	% Time Meet Requirements	95%

### 4.2.3 GINA Step 3 – Develop System Metrics

To compare all the different architectures formulated during the conceptual design phase, GINA uses a quantifiable set of metrics – capability, performance, lifecycle cost, cost per function, and adaptability. The capability metrics have already been introduced in Section 4.2.2. The remaining four metrics are discussed below. These are the metrics that will be used to evaluate and compare alternative TPF architectures.

## Performance

While the four capability quality of service metrics measure how well the architecture meets the capability requirements at any *instantaneous* point in time, the performance metric measures how well the architecture satisfies the demands of the market over the *entire* mission life [Shaw, 1998]. Because the imaging rate was identified as the distinguishing capability metric between TPF architectures, performance is expressed as the total number of images the system produces over the mission duration. In this case, "images" refers to the sum of all surveys, medium spectroscopies, and deep spectroscopies.

To calculate this quantity, however, the additional complication of taking into account all of the possible failures that may occur within the system must be addressed. As individual components within TPF fail over time, the imaging rate of the system will decrease in order to maintain the same pre-failure level of integrity (i.e. TPF must still achieve the same SNR for each operational mode, despite failures). Thus, the imaging rate capability metric is a dynamic quantity that changes over time.

To take into account potential failures and the effect they have on system performance, a reliability model is needed. GINA uses Markov reliability modeling techniques to determine both the probability that the system will continue to function over a given amount of time and the likelihood with which the system will function in different partially failed states throughout the mission. A Markov model must be created for each proposed architecture. From the Markov model, a system of differential equations is developed, as shown in Eqn. 4.1, where  $\dot{\vec{P}}$  is the time rate of change of the state probability vector,  $A$  is the state coefficient matrix comprised of component failure rates, and  $\vec{P}$  is the state probability vector.

$$\dot{\vec{P}} = A\vec{P} \quad (4.1)$$

The solution to this set of linear, first-order, partially coupled ordinary differential equations determines the probability of the TPF system being in any given state at a particular time. Coupling the outputs of the reliability model with the outputs of the capability model (i.e. the imaging rate of the TPF architecture in each operational state) yields the total performance (i.e. total number of images) of the system. The coupling equation is the utility function:

$$\Psi(T) = \int_0^T \sum_{i=1}^n C_i P_i(t) dt \quad (4.2)$$

where  $\Psi$  is the total number of images produced,  $T$  is the mission lifetime,  $n$  is the total number of operational states (ex. state 1 = 8 out of 8 collector apertures are functional, state 2 = 7 out of 8 collector apertures are operational, etc.),  $C_i$  is the capability (i.e. imaging rate) in each state  $i$ , and  $P_i(t)$  is the probability of being in each operational state  $i$  as a function of time  $t$ .

### Lifecycle Cost

Four primary components of the lifecycle cost model are the payload cost, satellite bus cost, launch cost, and operations cost. Payload and satellite bus theoretical first unit (TFU) costs are computed via a NASA Goddard cost estimation relationship (CER) for infrared apertures and the Air Force Unmanned Spacecraft Cost Model cost estimation relationships in Table 3.3. Then a learning curve is applied to compute the total cost in constant year dollars as a function of the required number of units for a particular system architecture. The launch cost model uses the optimization algorithm in Appendix B to select a launch vehicle(s) from the complete set being considered by NASA to deploy TPF at minimum cost. Inputs to the launch cost model are the total number of spacecraft in the TPF architecture, the mass and volume of the spacecraft, the heliocentric orbital radius of the spacecraft, and the performance parameters of each potential launch vehicle. Finally, the operations cost model estimates the operations crew size, operations crew cost, flight/

ground software size, software costs, and total operations lifecycle cost of operating the TPF spacecraft over the mission duration.

### **Cost Per Function**

The cost per function (CPF) metric provides a clear measure of the cost of an architecture versus its performance. It is a measure of the cost to achieve a common level of performance, and includes expected development, launch, failure compensation, and operations costs [Shaw et al, 2001]. For TPF, the cost per function is defined as the cost per image, and is calculated by dividing the total lifecycle cost of a TPF architecture by the total number of images the architecture produces over its mission duration.

### **Adaptability**

Adaptability measures how flexible an architecture is to changes in design assumptions and mission requirements [Saleh, 2002]. In one sense, adaptability may be thought of as the sensitivity or elasticity of the CPF of a particular architecture to incremental changes in an assumption or requirement. For the TPF mission, potential assumptions that could be altered to measure architecture sensitivity include component costs, learning curve slopes, and component reliabilities. In another sense, adaptability may be thought of as the flexibility of a particular architecture to a new set of mission requirements. An example of flexibility for TPF might be the ability of an architecture to transition from a planetary detection mission to a mission of astrophysical imaging.

## **4.2.4 GINA Step 4 – Partition the Conceptual Design Problem**

The next step in the GINA process involves listing all of the design variables the systems engineer has control over. From this list, key independent variables that drive the design of the system are extracted and inserted into the design vector. The design vector contains the parameters that will distinguish and differentiate alternative system architectures. Remaining variables are placed in the constants vector. The constants vector contains parameters the systems engineer has control over, but that will not distinguish or differen-

tiate alternative architectures. For example, the assumed areal density of the satellite solar arrays is set to a constant value as a function of the type of solar array (ex.  $1 \text{ kg/m}^2$  for gallium arsenide multi-junction cells) regardless of the other attributes of the architecture.

For TPF, four variables – heliocentric orbital radius, collector connectivity/geometry, number of collector apertures, and collector aperture diameter – make up the TPF design vector. Table 4.2 lists the range of values considered for each design vector variable in the TPF case study. Each design vector defines a unique TPF system architecture.

**TABLE 4.2** The TPF Design Vector

Design Vector Variable	$\Gamma$	Range	Discretization
Heliocentric Orbital Radius	$\gamma_1$	1.0 AU - 5.5 AU	0.5 AU
Collector Connectivity/Geometry*	$\gamma_2$	SCI-1D, SCI-2D, SSI-1D, SSI-2D	Unique
# Collector Apertures	$\gamma_3$	4 - 10	2
Collector Aperture Diameter	$\gamma_4$	1 m - 4 m	1 m
<b>Total # Permutations</b>		<b>640</b>	

\*SCI, structurally connected interferometer; SSI, separated spacecraft interferometer; 1D, one-dimensional array; 2D, two-dimensional array

Once the design vector has been identified, it may be matrixed against the GINA system metrics. The chart in Figure 4.3 establishes the relationship between the trade space for TPF and the metrics by which competing architectures are judged. Notice the fundamental relationships between the elements of the design vector and the GINA metrics. For example, the number of apertures in the system directly affects the ability to shape the transmissivity function. This dictates the sharpness in the rise of the transmissivity at the boundary between the exo-zodi and the habitable zone. Hence, the number of apertures drives the isolation metric (i.e. null depth and angular resolution).

Once this matrix has been completed, different attributes can be lumped into groups of modeling needs to enable the recognition of important differences between competing architectures. The systems engineer may now define the modules that will be required to

<b>Top Trades Capability Metrics</b>	Heliocentric Orbital Radius (1 to 6 AU)	Aperture Maintenance (SCI vs. SSI)	Number of Apertures (4 to 12)	Size of Apertures (1 to 4 m)
Isolation (Angular Res.)	N/A	SSI allow more freedom in baseline tuning	Fine tuning of transmissivity function	N/A
Rate (Images/Life)	Noise reductions increase rates. Different operation delays	SSI power and propulsion requirements highly sensitive	Increased collecting area improves rate	Increased collecting area improves rate
Integrity (SNR)	Different local zodiacal emission and solar thermal flux	SCI: passive alignment but complex flexible dynamics	Tuning of transmissivity for exo-zodiacal suppression	Smaller FOV collects less local zodiacal noise
Availability (Variability)	N/A	Different safing complexity and operational events	Different calibration and capture complexity	N/A

Aperture

Systems Analysis

Operations

Dyn & Controls

Environment

S/C

**Figure 4.3** Terrestrial Planet Finder Model Component Identification Metrics Matrix [Curtis et al, 1999]

develop, analyze, and compare different architectures for the TPF mission. These modules capture the fundamental physics, costs, and systems engineering trades of the TPF conceptual design problem. For TPF, six core modules are required:

1. Environment module
2. Aperture Configuration module
3. Spacecraft Payload and Bus module
4. Dynamics, Control, & Stability module
5. Deployment & Operations module
6. Systems Analysis module

Detailed analysis of the model component identification metrics matrix begins to yield insight to the trade space. Figure 4.4 illustrates an alternative view of the TPF trade space. Each blue arrow represents an element of the design vector, while the green and

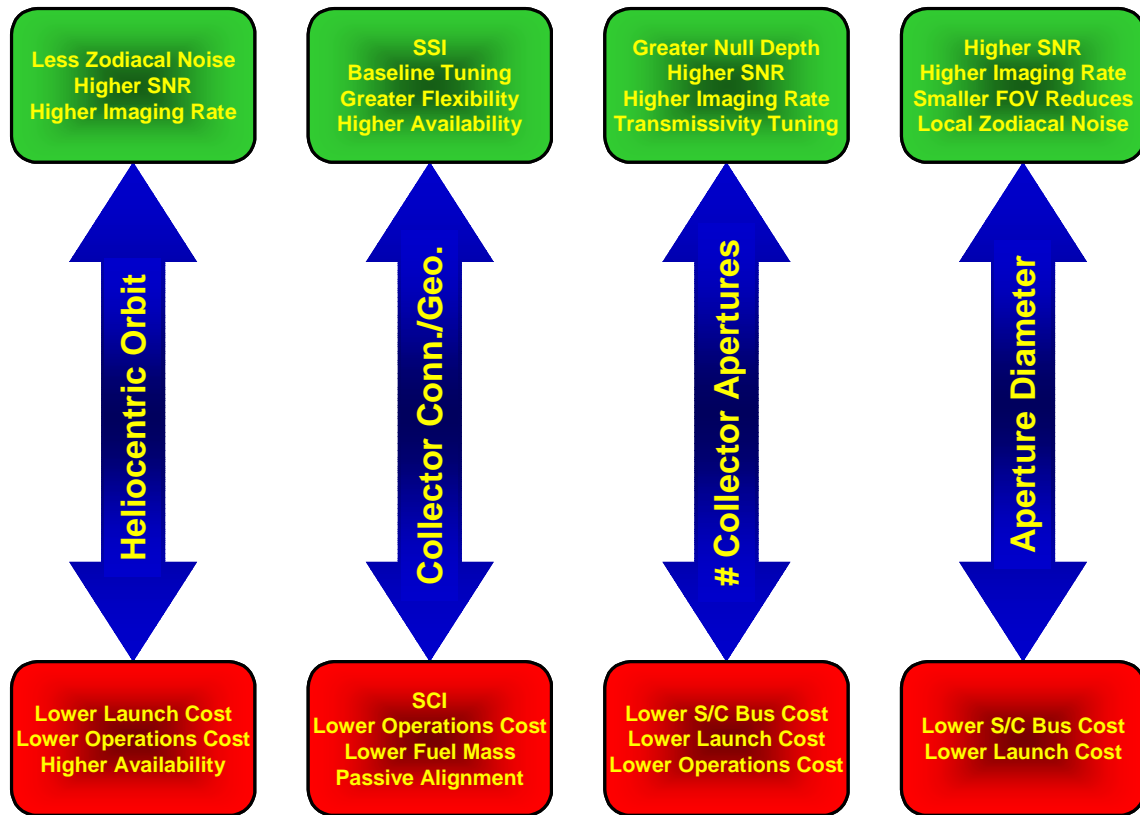


Figure 4.4 Alternative View of the TPF Trade Space

red boxes represent the benefit in terms of the GINA metrics of increasing or decreasing the value of that parameter. For example, increasing the heliocentric orbital radius improves the integrity and rate capabilities of the system, but at the expense of launch costs, operations costs, and system availability. Likewise, decreasing the total number of collector apertures lowers the lifecycle cost, but at the expense of the null depth, transmissivity tuning, signal-to-noise ratio, and imaging rate. Similar trade-offs exist for each element of the design vector.

What makes the TPF conceptual design problem so difficult is that all of these trades are coupled in a nonlinear fashion. Each element of the design vector cannot simply be optimized in isolation from each other, because each element of the design vector impacts all elements of the design. This is where the true strength of the MMDOSA methodology demonstrates itself, as MMDOSA enables the systems engineer to capture quantitatively



all these couplings and then search the nonconvex trade space for the most cost-effective families of system architectures.

#### 4.2.5 GINA Step 5 – Develop Simulation Software

This section provides a high-level overview of the TPF GINA model as excerpted from the references by Curtis et al [Curtis et al, 1999] and Miller et al [Miller et al, 2000]. For greater detail on the TPF GINA model, the assumptions behind the model, and the analytical relationships inside the model; please refer to these two references.

A module is a Matlab m-file that models a specific subsystem or function within the mission. The purpose of the modules is to develop a tool that sizes the subsystems within an architecture for a given set of inputs. If implemented correctly, the modules should also aid the systems analysis methodology by computing the metrics for a given architecture. This allows for rapid evaluations and comparisons of alternative TPF architectures.

Figure 4.5 illustrates the Matlab modules identified in the model component identification metrics matrix within the context of the TPF GINA model. The two sets of inputs to the model are the design vector and constants vector. First, the environment module computes properties such as solar flux and gravity gradient local to the selected TPF orbit. Next, the aperture configuration module determines the optimal collector aperture geometry that satisfies both the nulling requirements and the architecture constraints. The spacecraft payload and bus module then designs and sizes all of the spacecraft. This information feeds into the dynamics, control, and stability module, which creates a finite element model of the spacecraft, designs an optical control system, and performs a disturbance analysis to ensure that optical path delay tolerances are met. Next, the launch and operations module computes the orbital transfer parameters, selects a launch vehicle, assesses operations complexity, and estimates the lifecycle operations cost. Finally, the systems analysis module determines the system's ideal imaging rate in each operational mode, estimates the total number of images obtained over the mission duration taking into account mission inefficiencies and spacecraft failures, and calculates the lifecycle cost.

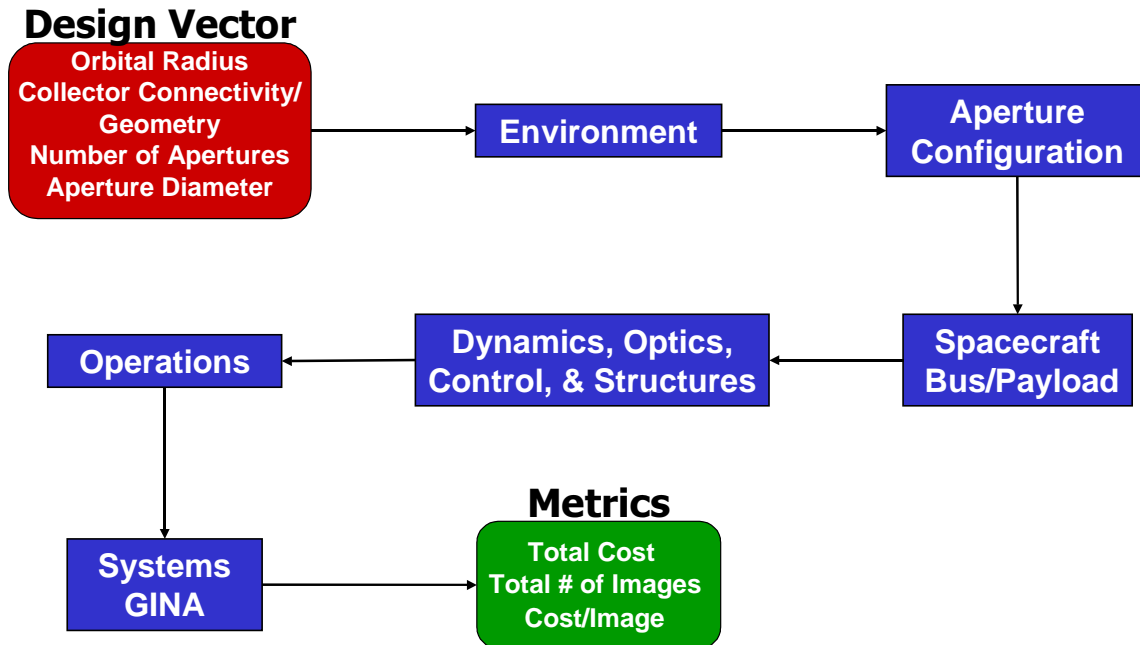


Figure 4.5 TPF GINA Model Simulation Software Block Diagram

The mathematical models within each module can become intricate, making the consistent management of information within the model a complex issue. An  $N^2$  diagram is used as the interface control document during this step. Figure 4.6 illustrates the TPF GINA model  $N^2$  diagram. As one can see, certain modules are further decomposed into submodules. This modular division of the TPF conceptual design process reduces software development risk by reducing coupling and simplifies the simulation code development as each module is separately testable.

An  $N^2$  diagram, also known as a Design Structure Matrix [Browning, 1998], is an  $N \times N$  matrix used by systems engineers to develop and organize interface information [Kockler, F.R., 1990]. The submodules (i.e. Matlab m-file functions) are located along the diagonal of the matrix. Each color-coded number designates a unique variable within the submodule. The inputs to each submodule are vertical and the outputs are horizontal. The aggregation of the submodules into complete modules is illustrated by the black boxes enveloping different sections of the diagonal. The  $N^2$ -diagram in Figure 4.6 provides a visual representation of the flow of information through the conceptual design process and

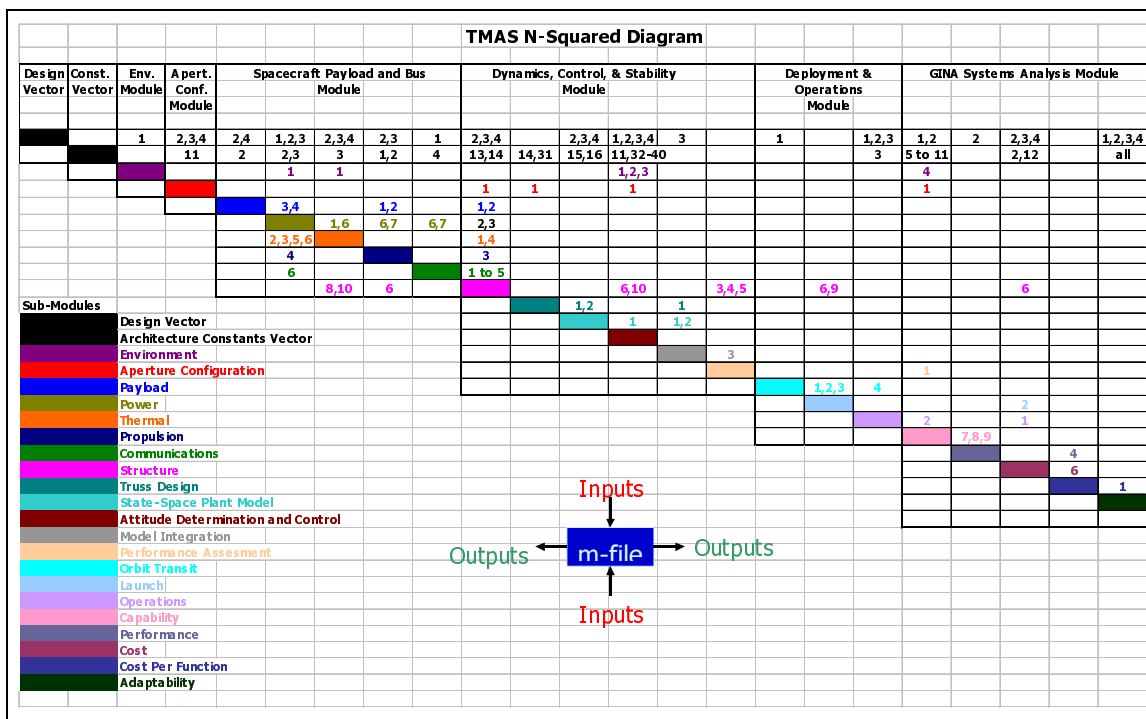


Figure 4.6 N<sup>2</sup>-Diagram for the TPF GINA Model

is used to connect all of the Matlab functions to enable an automated simulation of different TPF architectures.

### Environment Module

Figure 4.7 illustrates key inputs and outputs of the environment module. With knowledge of the heliocentric orbital radius from the design vector, the local space environment around the TPF spacecraft in its operational orbit is characterized. Key outputs include solar flux, local zodiacal intensity, solar pressure, and gravity gradient.

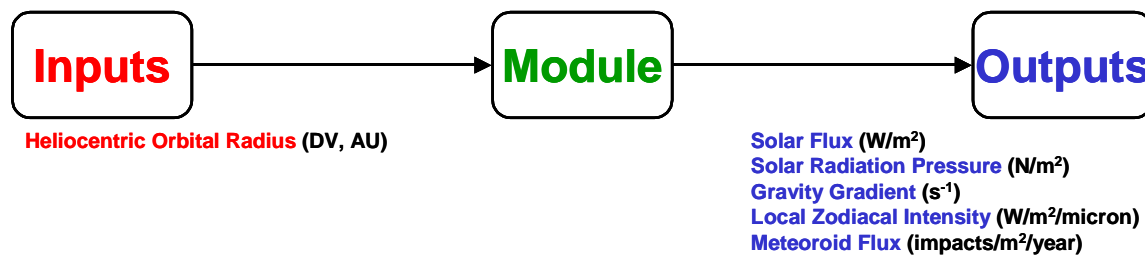


Figure 4.7 Environment Module Inputs and Outputs

The solar flux  $G_s$  determines both the type and size of power system and thermal control system, and may be calculated as:

$$G_s = \sigma T_s^4 \frac{R_s^2}{R_o^2} \quad (4.3)$$

where  $\sigma$  is the Stefan-Boltzmann constant,  $T_s$  is the temperature of the sun,  $R_s$  is the radius of the sun, and  $R_o$  is the heliocentric orbital radius of the TPF spacecraft. The local zodiacal cloud is comprised of tiny particles of dust that orbit the sun. These dust particles create a diffuse glow at infrared wavelengths and act as a noise source that affects the signal integrity and imaging rate of the TPF system. Intensity  $W_{T\lambda}$  of the local zodiacal dust cloud may be computed as:

$$\begin{aligned} W_{T\lambda} &= RE(T_{LZ}, \lambda) \tau_{LZ} 10^{-6} \\ RE(T_{LZ}, \lambda) &= 2\pi c^2 \frac{h}{\lambda^5 \left( e^{\frac{hc}{\lambda T_{LZ} k}} - 1 \right)} \\ T_{LZ} &= 265 L_s^{0.25} R_o^{-0.4} \\ \tau_{LZ} &= 6 \times 10^{-8} R_o^{-0.8} \end{aligned} \quad (4.4)$$

where  $T_{LZ}$  is the local zodiacal temperature,  $\tau_{LZ}$  is the local zodiacal optical depth,  $\lambda$  is the wavelength of observation,  $h$  is Planck's constant,  $c$  is the speed of light,  $k$  is Boltzmann's constant, and  $L_s$  is the solar luminosity.

### Aperture Configuration Module

Figure 4.8 illustrates key inputs and outputs of the aperture configuration module. With knowledge of the collector connectivity/geometry, the number and diameter of collector apertures, and the observational wavelength of the interferometer; the optical layout of the interferometer is determined. Key outputs include the geometric configuration of the collector apertures and the delay phasing for nulling.

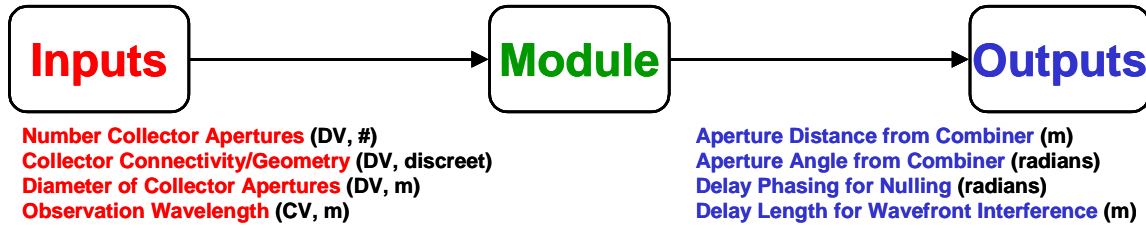


Figure 4.8 Aperture Configuration Module Inputs and Outputs

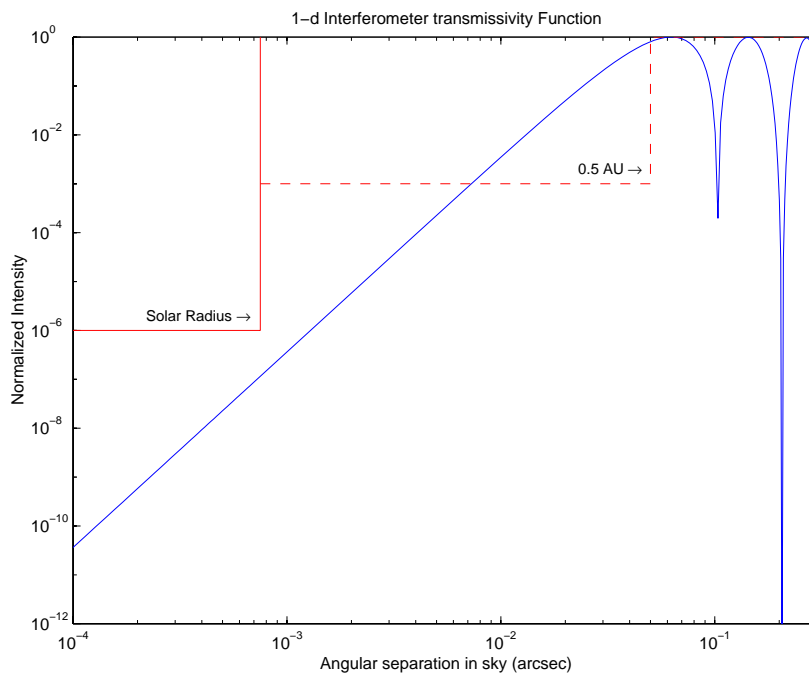
The TPF mission will be implemented using a number of apertures operating as an interferometer. Unlike conventional interferometers, the goal of this mission is to null out the parent star at the center of the field of view. This will be accomplished using the Bracewell nulling interferometer concept [Bracewell & Macphie, 1979]. However, similar to any conventional interferometer, the response of the TPF interferometer will be highly dependent upon the locations and the relative sizes of the individual apertures. This module takes into account the external noise sources that affect the capability of the interferometer to detect a planet and calculates the optimal transmissivity function for the given number of apertures. The aperture configurations were optimized for imaging an Earth-Sun system located 10 parsecs away at an observational wavelength of 12  $\mu\text{m}$ .

The nulling interference pattern communicated as the transmissivity function is given by:

$$\Theta = \left| \sum_{k=1}^N D_k \exp \left[ j2\pi \left( \frac{L_k r}{\lambda} \right) \cos(\delta_k - \theta) \right] \exp(j\phi_k) \right|^2 \quad (4.5)$$

where  $N$  is the number of apertures in the array,  $D_k$  is the diameter of aperture  $k$ ,  $L_k$  is the distance between aperture  $k$  and the center of the array,  $r$  is the angular separation of the source from the center of the interferometer's fringe pattern,  $\lambda$  is the wavelength of observation,  $\delta_k$  is the clock angle of aperture  $k$  measured from a given aperture,  $\theta$  is the azimuth angle of the source from the center of the interferometer's fringe pattern, and  $\phi_k$  is the independent phase shift introduced to beam  $k$ . Figure 4.9 shows the transmissivity function for an example interferometer [Mennesson & Mariotti, 1997]. The response shown here meets the nulling requirement to detect an Earth-like planet orbiting a Sun-like star

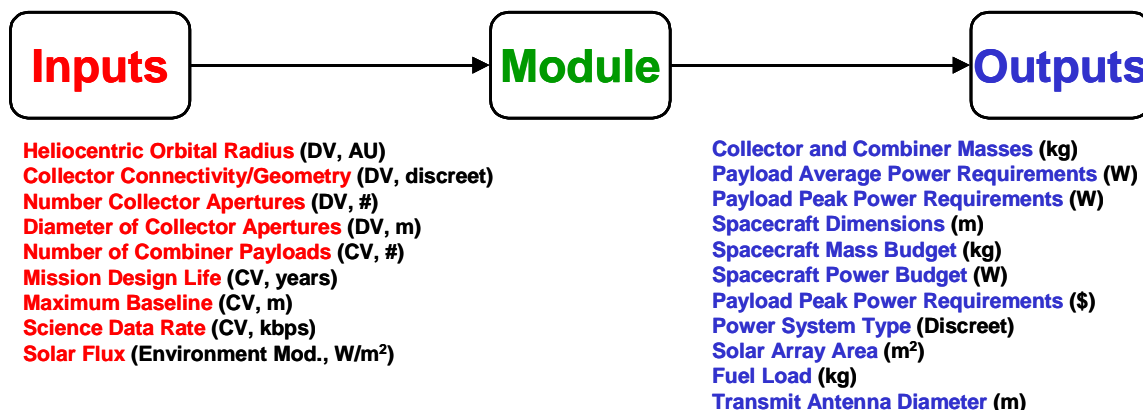
located 10 parsecs away. The six order-of-magnitude star light suppression requirement is indicated by the solid box in the figure. The dashed line in the figure represents an area between the surface of the parent star to the 0.5 AU inner limit of the habitable zone. Signals received from this area will be predominantly from the exo-zodiacal cloud surrounding the star. Therefore it is desirable to also null out this region. Unfortunately, it is almost impossible to null out such a large region using a limited number of apertures while maintaining the desired high transmissivity in the habitable zone region. Hence, the dashed lines represent a soft constraint where it is preferable to have the interferometer exhibit a low transmissivity.



**Figure 4.9** Sample TPF Transmissivity Function [Curtis et al, 1999]

### Spacecraft Payload and Bus Module

Figure 4.10 lists key inputs and outputs of the spacecraft payload and bus module. With knowledge of the design vector inputs in combination with the mission duration, maximum baseline, science data downlink rate, and solar flux; the payload is sized to support



**Figure 4.10** Spacecraft Payload and Bus Module Inputs and Outputs

the collectors and combiner, and then the spacecraft is sized to support the payload. Key outputs include the dimensions, mass budget, and power budget of the spacecraft.

The spacecraft payload and bus module consists of five separate submodules that model the payload, communications, power, propulsion, and thermal subsystems. The payload submodule models four primary components of the TPF payload: collector mirrors, optical train, beam combiner, and infrared detectors. The communications submodule calculates the minimum antenna size and mass for a given power, data rate, and heliocentric orbital radius. Outputs from the communications submodule are the antenna diameter, antenna mass, transmitter mass, and minimum gimbal distance. The power submodule chooses the lowest mass power source from the two power sources used in modern spacecraft – solar arrays and radioisotope thermoelectric generators (RTGs). Power system masses are estimated using average and peak power requirements for the bus and payload subsystems and the mission duration, interferometer type, and orbital radius. Solar arrays, RTG's, and batteries are all sized in this submodule. The propulsion submodule chooses an appropriate attitude control thruster design. The propulsion system provides energy to maneuver the TPF spacecraft between targets and to dump momentum. In a separated spacecraft architecture, the propulsion system must also provide the energy to rotate each collector spacecraft about the combiner, maintaining the spacecraft in constant relative positions. In this case, the propulsion submodule is biased to design a system that can sus-

tain continuous low thrust levels. Finally, the thermal submodule estimates the mass and power of the spacecraft thermal control system to maintain the infrared detector, optical train components, and spacecraft electronics at the required cold temperatures. The infrared detector and optical train temperature requirements are derived from the specifications for the instrument sensitivity in the near infrared frequencies. Since the ambient black-body temperature for the spacecraft is expected to be between 100 Kelvin and 200 Kelvin for the range of solar orbits under consideration, both cooling and heating capabilities for different parts of the spacecraft will be needed. Using a combined strategy of thermal shields, cryocoolers, and heaters, this module chooses the lowest mass option that satisfies the thermal requirements.

### **Dynamics, Control, and Stability Module**

Figure 4.11 illustrates key inputs and outputs of the dynamics, control, and stability module. This module uses information from the design and constants vectors in combination with outputs from all of the preceding modules to model the difficulties in maintaining nanometer precision optical path length control between the apertures and combiner in the TPF interferometer. Key outputs include a finite element model of the system, identification of the natural frequencies of the system, and measurement of the dynamic noise optical pathlength differences. This module contains five submodules – the structures, optics control, attitude determination and control system, integration, and disturbance analysis submodules.

The structures submodule creates dynamic models for the structurally connected (SCI) and separated spacecraft (SSI) interferometer architectures. It uses a generic spacecraft design based on the concept of a central hub, which contains the combiner, spacecraft bus, and high-gain communications antenna. The apertures are located in a plane around the hub. In the structurally connected case, a deployable truss connects each aperture with the central hub or with another aperture that is located on the same radial spoke. Each truss and its associated canister are dimensioned based on existing empirical engineering relationships for truss diameter and mass. Using finite element modeling methods and matrix



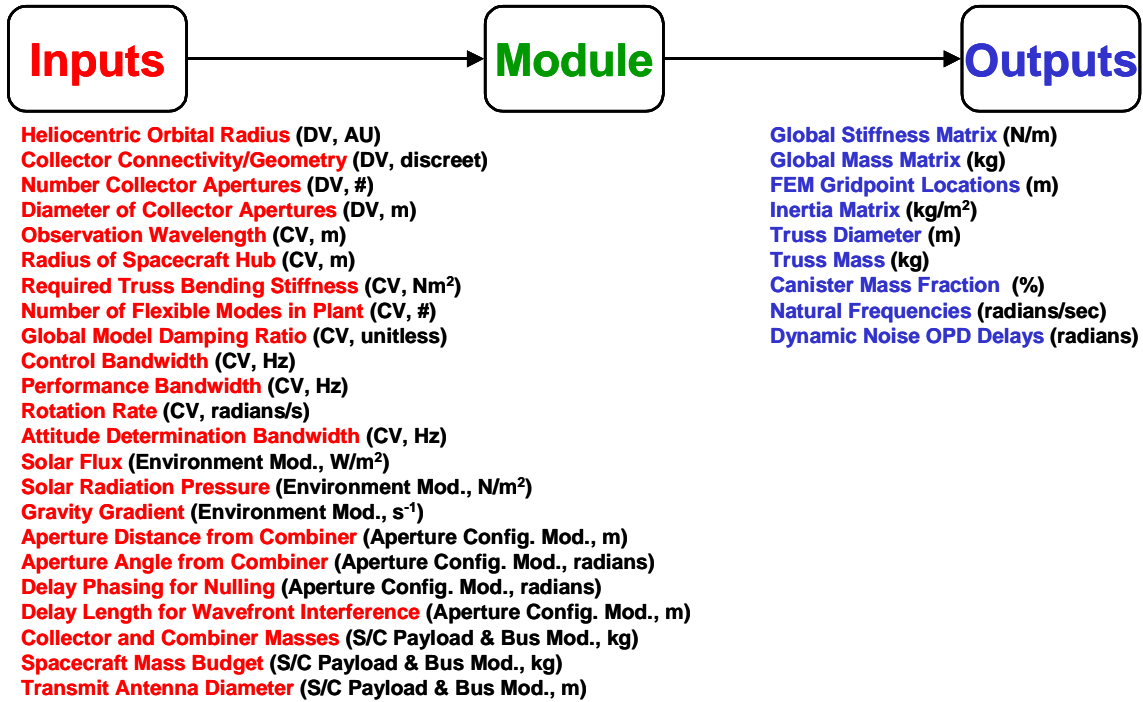
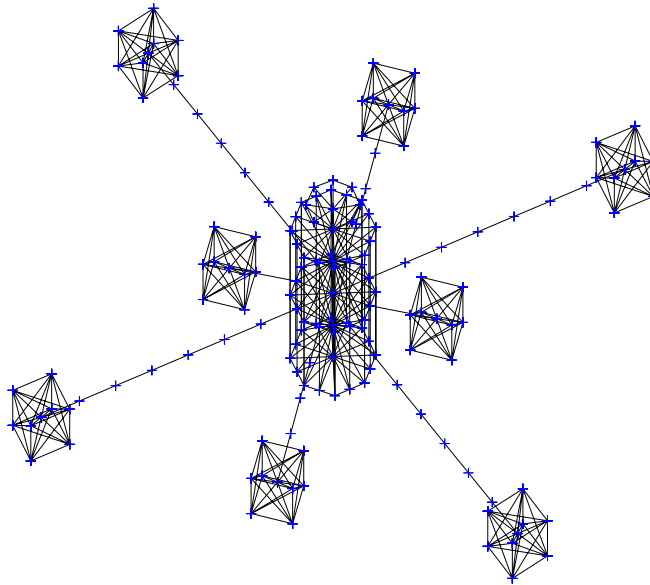


Figure 4.11 Dynamics, Control, and Stability Module Inputs and Outputs

algebra, the structural mass, mass distribution, and approximate dynamic characteristics are calculated. Figure 4.12 shows a graphical representation of a TPF finite element model for a structurally connected, eight two-meter aperture two-dimensional array.

The optics control submodule calculates the optics linear sensitivity matrix, which relates the physical displacements and rotations of the combiner and the apertures to the optical performance metrics. The optical performance metric is the optical pathlength difference (OPD), which can be calculated as a linear combination of the relative displacements of the hub and apertures with respect to each other as shown in the following sequence of relationships:

$$\begin{aligned}
 \text{OPL}_{ref} &= R - z_{ref} + r_{ref} + (\bar{d}_{ref}\bar{u}_{ref}) - (\bar{d}_o\bar{u}_{ref}) \\
 \text{OPL}_i &= R - z_i + r_i + r_{ODLi} + (\bar{d}_i\bar{u}_i) - (\bar{d}_o\bar{u}_i) \\
 \text{OPD}_i &= \text{OPL}_{ref} - \text{OPL}_i
 \end{aligned}
 \tag{4.6}$$



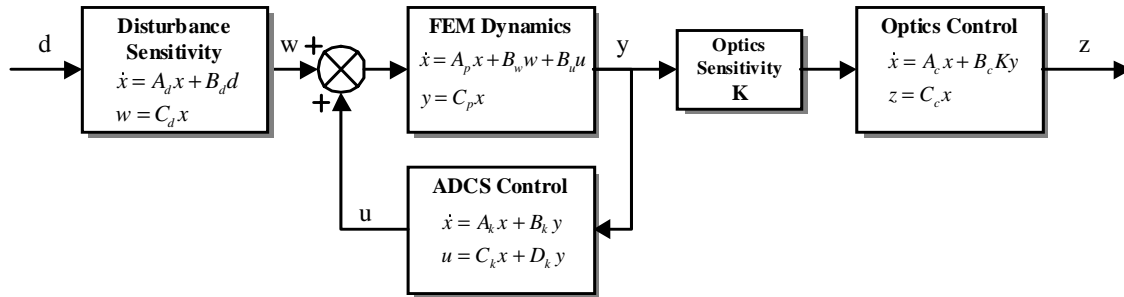
**Figure 4.12** Example TPF Finite Element Model

where OPL is optical pathlength,  $R$  is the distance of the aperture to the reference plane of the incoming stellar wavefront,  $z$  is the position of the aperture  $i$ ,  $r$  is the radial position of the aperture,  $d$  is the displacement vector of the aperture, and  $u$  is the unit normal vector along the aperture radius.

To provide the nulling performance needed for planet detection, the relative geometry of the TPF apertures and other optical instruments must be maintained within very small tolerances. To satisfy this requirement and to allow for the limited bandwidth and dynamic range of available sensors and actuators, a layered control system is employed. While the optics control submodule is responsible for the final nanometer scale control, the attitude determination and control system (ADCS) submodule provides the coarser, centimeter scale control between apertures. The primary purpose of the ADCS is to stabilize rigid body motions and reject disturbances at low frequencies. This is achieved by sensing the current attitude, and relative position in the SSI case, using an appropriate suite of sensors and providing control torques and forces using thrusters and angular momentum control/

storage devices. The ADCS submodule characterizes the achievable performance in the face of the worst-case environmental disturbances and the spacecraft disturbances injected into the system by the ADCS actuators themselves – primarily the effects of static and dynamic imbalances in the reaction wheel assemblies.

The integration submodule assembles an integrated dynamic spacecraft model based on the outputs from the dynamics, control, and stability module and the ADCS submodule. The integrated model is represented in LTI state space form and can be used for the subsequent dynamic performance analysis. The integrated model contains the structural plant, ADCS, optical controllers, and linear sensitivity matrix that relates the physical degrees of freedom of the structural system to the performance metrics of interest as illustrated in Figure 4.13.



**Figure 4.13** Integrated Dynamics Model Block Diagram [Curtis et al, 1999]

Finally, the disturbance analysis sub-module models disturbances as power spectral densities and evaluates their effect using the integrated spacecraft dynamic model derived in the integration submodule. The RMS phase error  $\sigma_z$  caused by a disturbance can then be calculated in the following manner:

$$\sigma_z^2 = \frac{1}{\pi} \int_0^{\infty} G_{zw}(\omega) S_{ww}(\omega) G_{zw}^H(\omega) d\omega \tag{4.7}$$

where  $G_{zw}$  is the transfer function from disturbance to performance output and  $S_{ww}$  is the disturbance cross spectral density matrix,.

## Launch and Operations Module

Figure 4.14 illustrates key inputs and outputs of the launch and operations module. With knowledge of the heliocentric orbital radius, mission duration, spacecraft mass, and spacecraft dimensions; the launch and operations requirements for TPF are estimated. Key outputs from this module include launch vehicle selection, operations crew size, launch cost, and lifecycle operations cost. This module is divided into three submodules – the orbital transit, launch and operations submodules.

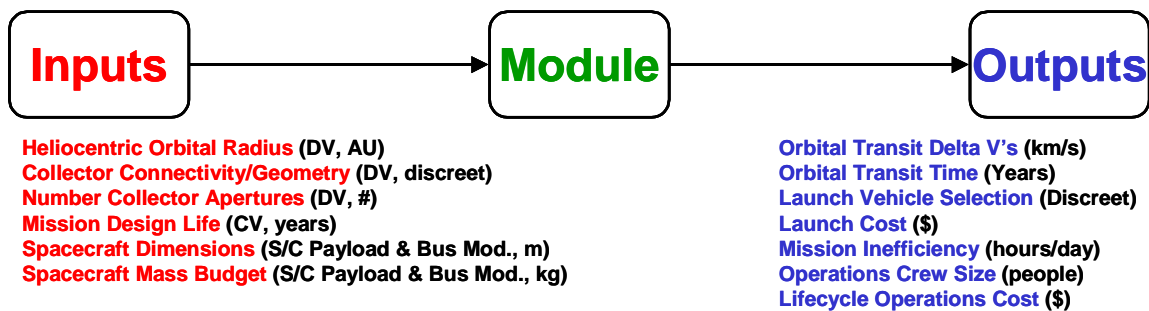
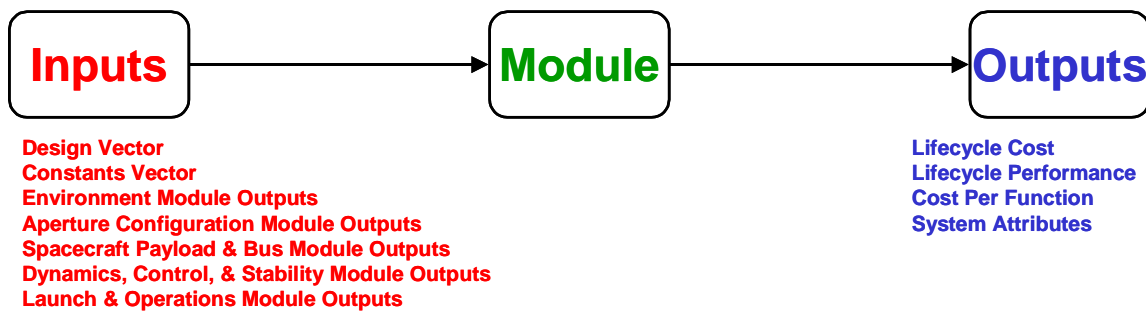


Figure 4.14 Launch and Operations Module Inputs and Outputs

The orbital transit submodule performs  $\Delta V$  and flight time calculations for a Hohmann's transfer [Hale, 1994] to the desired orbit. The integer programming approach presented in Appendix B is used to solve the TPF launch vehicle selection problem as an optimization problem. The decision variables in the integer programming formulation of the launch vehicle selection problem are how many of each of the launch vehicles under consideration – the Zenit, Delta II, Delta III, Delta IV, Ariane V, and Titan IV – should be used to deploy the system. The objective function is to select a launch vehicle(s) to deploy TPF at minimum cost. Finally, the operations submodule estimates the operational complexity of the TPF architecture as a function of the interferometer type, and the mission inefficiency as a function of both the heliocentric orbital radius and the system reliability. This information in turn determines the size of the operations crew and operations software, which in turn is used to estimate the lifecycle operations cost of the TPF architecture.

## Systems Analysis Module

Figure 4.15 illustrates inputs and outputs of the systems analysis module. With knowledge of the outputs from all the preceding modules in the TPF GINA model, key system attributes that will be used to compare various conceptual design architectures may be computed. Key outputs include the lifecycle cost, lifecycle performance, and cost per function.



**Figure 4.15** Systems Analysis Module Inputs and Outputs

The total lifecycle cost  $\Phi_y$  over five years of a TPF architecture is computed as:

$$\sum_{y=1}^5 \Phi_y = C_{PL} + C_{SC} + C_L + C_{OPS} \quad (4.8)$$

where  $y$  is the year in the mission,  $C_{PL}$  is the payload cost,  $C_{SC}$  is the spacecraft bus cost,  $C_L$  is the launch cost, and  $C_{OPS}$  is the operations cost. The theoretical first unit payload cost is computed by a NASA Goddard cost estimation relationship that scales payload cost with aperture diameter, and the theoretical first unit spacecraft bus cost is computed via the U.S. Air Force Unmanned Spacecraft Cost Model relationships in Table 3.3. A learning curve is then applied as a function of the number of payloads and spacecraft in the TPF architecture to compute  $C_{PL}$  and  $C_{SC}$ . Both the launch and operations costs are computed in the launch and operations module.

System capability is a measure of the imaging rate of a TPF architecture. The minimum integration time  $\tau$  required for observing a planet is:

$$\tau = \left[ \frac{\text{SNR} \sqrt{Q_{\text{leak}} + Q_{\text{LZ}} + Q_{\text{EZ}} + Q_{\text{dark}} + Q_{\text{planet}} + Q_{\text{flat}}}}{Q_{\text{planet}}} \right]^2 \quad (4.9)$$

where SNR is the required signal-to-noise ratio for the current mode of operation (i.e. surveying, medium spectroscopy, or deep spectroscopy),  $Q_{\text{leak}}$  is the leakage signal,  $Q_{\text{LZ}}$  is the local zodiacal signal,  $Q_{\text{EZ}}$  is the exozodiacal signal,  $Q_{\text{dark}}$  is the detector noise,  $Q_{\text{flat}}$  is the background noise, and  $Q_{\text{planet}}$  is the planet signal modulated by the fringe pattern as the interferometer rotates around the line of sight to the star. The image rate  $I$  is the inverse of the sum of the integration time  $\tau$  and overhead time  $T_o$ , where overhead includes the time allotted for interferometer slewing, aperture configuration, array rotation, and mission operations inefficiency:

$$I = (\tau + T_o)^{-1} \quad (4.10)$$

Note that each TPF architecture will have three separate imaging rates as the required SNR for surveys, medium spectroscopies, and deep spectroscopies varies from 5 to 25.

Mission performance  $\Psi$  measures the total number of images the system obtains over five years, taking into account the degradation in system capability as partial failures occur over time. An example of system degradation would be the decrease in imaging rate that would result from the failure of one or more collector apertures in the interferometer. Markov reliability modeling techniques are used to determine both the probability that the system will continue to function over a given amount of time and the likelihood with which the system will function in different partially failed states throughout the mission. A Markov model must be created for each proposed architecture. From the Markov model, a system of differential equations is developed, as shown in Eqn. 4.11, where  $\dot{\vec{P}}$  is the time rate of change of the state probability vector,  $A$  is the state coefficient matrix comprised of component failure rates, and  $\vec{P}$  is the state probability vector.

$$\dot{\vec{P}} = A\vec{P} \quad (4.11)$$

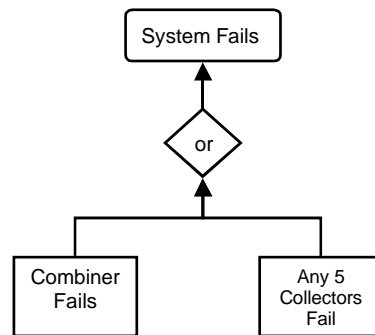
The solution to this set of linear, first-order, partially coupled ordinary differential equations determines the probability of TPF being in any given state at a particular time. Coupling the outputs of the reliability model with the outputs of the capability model (i.e. the imaging rate of the TPF architecture in each operational state) yields the total performance (i.e. total number of images) of the system. The coupling equation is the utility function. A generalized utility function that may be applied to compute the performance of any DSS is given in Eqn. 4.2.

$$U(T) = \int_0^T \sum_{i=1}^n C_i P_i(t) dt \quad (4.12)$$

$U$  is the total performance (i.e. number of images for TPF),  $T$  is the mission lifetime,  $n$  is the number of operational states,  $C$  is the capability in each state  $i$  (i.e. imaging rate for TPF), and  $P$  is the probability of being in each operational state  $i$  as a function of time  $t$ .

Take the example of an eight-aperture TPF architecture. To apply a Markov modeling methodology, the states of the system must be time dependent, sequential, and mutually exclusive. If the system satisfies these requirements, then a set of differential equations can be written to model the evolution of the system. By taking advantage of the Markov property, which states that given full knowledge of the current state of the system, one can predict all future states by integrating the set of differential equations, irrespective of the past states of the system; the probability of being in any given state of the system at any given time may be computed [Gelb, 1974].

The first step is to develop a fault tree diagram illustrating all the possible modes of failure for a design (Figure 4.16). The minimum functionality required for a nulling interferometer is one combiner and four collector apertures. Thus, the eight-collector TPF architecture fails when the combiner fails or when any five collectors fail. The system will still



**Figure 4.16** Aggregated Fault Tree for an Eight-Collector TPF Architecture

function when one to four collector apertures fail, but at a reduced capability (i.e. reduced imaging rate for TPF).

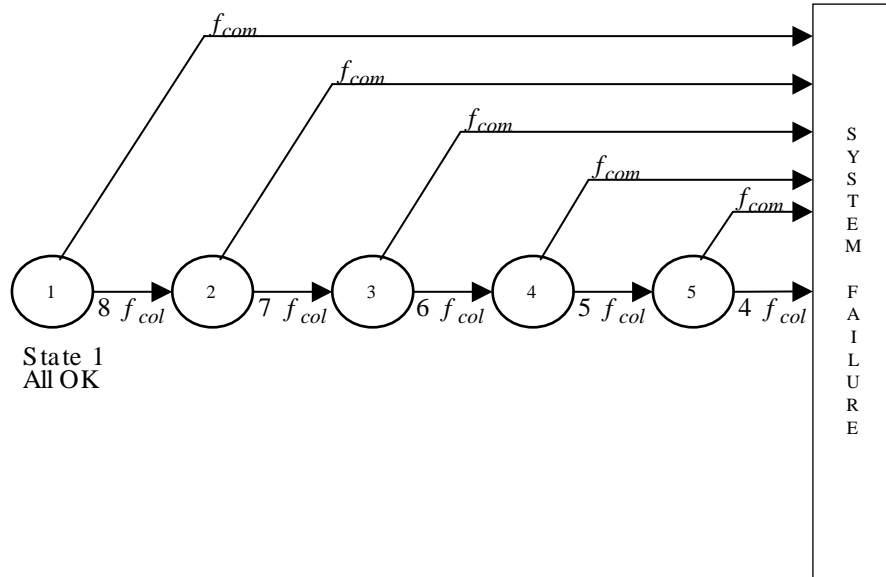
From the fault tree, a Markov model illustrating each possible state of the system may be created for each architecture. Figure 4.17 illustrates the Markov model for the eight-collector TPF architecture. This model contains five possible functioning states, all of which require a functional combiner.

- State 1: All eight collectors are working.
- State 2: Seven of the eight collectors are working.
- State 3: Six of the eight collectors are working.
- State 4: Five of the eight collectors are working.
- State 5: Four of the eight collectors are working.

Otherwise, the system is in a state of failure because the nulling (i.e. isolation) requirement can no longer be met.

From the Markov model state diagram, a system of differential equations can be written to determine the probability of the system being in any given state at any given time. This is done by representing each possible state of the system in the Markov model as a node in a network. To determine the differential equation for a particular state, the flow into and out of the node representing that state is balanced. The eight-collector TPF architecture





**Figure 4.17** Markov Model State Diagram for the 8-Collector TPF Architecture

requires a set of five partially coupled, linear, first-order differential equations to model the system (Eqn. 4.13).

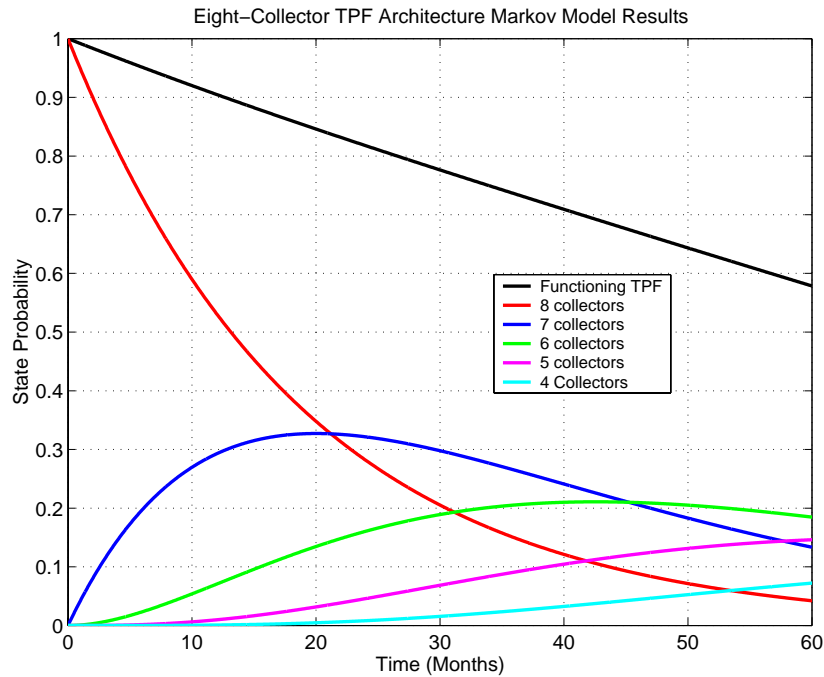
$$\begin{bmatrix} \dot{P}_1 \\ \dot{P}_2 \\ \dot{P}_3 \\ \dot{P}_4 \\ \dot{P}_5 \end{bmatrix} = \begin{bmatrix} -(f_{com} + 8f_{col}) & 0 & 0 & 0 & 0 \\ 8f_{col} & -(f_{com} + 7f_{col}) & 0 & 0 & 0 \\ 0 & 7f_{col} & -(f_{com} + 6f_{col}) & 0 & 0 \\ 0 & 0 & 6f_{col} & -(f_{com} + 5f_{col}) & 0 \\ 0 & 0 & 0 & 5f_{col} & -(f_{com} + 4f_{col}) \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \end{bmatrix} \quad (4.13)$$

Knowledge of the failure rates for the combiner ( $f_{com}$ ) and collector ( $f_{col}$ ) are required to solve for the state probabilities. In this example, it was assumed that a combiner has a mean-time-to-failure ( $mttf$ ) of 10 years and a collector has an  $mttf$  of 15 years. The failure rate  $f$  is the inverse of the  $mttf$ :

$$f = \frac{1}{mttf} \quad (4.14)$$

The system’s initial conditions are also required for the solution. In this case, the initial conditions (i.e. at  $t=0$  in the beginning of the operational mission) are a 100% probability

of being in state 1 and a 0% probability of being in all successive states. The plot in Figure 4.18 illustrates the results for the eight-collector TPF example. The Markov model gives the probability of being in any of the five operational states as a function of time through the five year (i.e. 60 month) TPF mission. Each of these five operational states will have a different survey, medium spectroscopy, and deep spectroscopy imaging rate.



**Figure 4.18** Markov Model Results for the Eight-Collector TPF Architecture ( $f_{com}=5.5 \times 10^{-3} \text{month}^{-1}$ ,  $f_{col}=8.3 \times 10^{-3} \text{month}^{-1}$ )

The solution to this system of differential equations (Eqn. 4.13) determines the probability  $P_i(t)$  of the system being in any given state  $i$  at a particular time  $t$ . Coupling, via the utility function in Eqn. 4.12, the outputs of the reliability model with the outputs of the capability model yields the total performance (i.e. total number of images  $\Psi$ ) of the system. The coupling equation is:

$$\begin{aligned}
\sum_{y=1}^5 \Psi_y &= \int_{74}^{256} \sum_{i=1}^n I_{si} P_i(t) dt + \int_{257}^{313} \sum_{i=1}^n I_{mi} P_i(t) dt + \int_{314}^{365} \sum_{i=1}^n I_{di} P_i(t) dt \\
&+ \int_{366}^{547} \sum_{i=1}^n I_{si} P_i(t) dt + \int_{548}^{658} \sum_{i=1}^n I_{mi} P_i(t) dt + \int_{659}^{730} \sum_{i=1}^n I_{di} P_i(t) dt \\
&+ \int_{731}^{804} \sum_{i=1}^n I_{si} P_i(t) dt + \int_{805}^{1004} \sum_{i=1}^n I_{mi} P_i(t) dt + \int_{1005}^{1095} \sum_{i=1}^n I_{di} P_i(t) dt \\
&+ \int_{1096}^{1132} \sum_{i=1}^n I_{si} P_i(t) dt + \int_{1133}^{1333} \sum_{i=1}^n I_{mi} P_i(t) dt + \int_{1334}^{1460} \sum_{i=1}^n I_{di} P_i(t) dt \\
&+ \int_{1461}^{1498} \sum_{i=1}^n I_{si} P_i(t) dt + \int_{1499}^{1680} \sum_{i=1}^n I_{mi} P_i(t) dt + \int_{1681}^{1825} \sum_{i=1}^n I_{di} P_i(t) dt
\end{aligned} \tag{4.15}$$

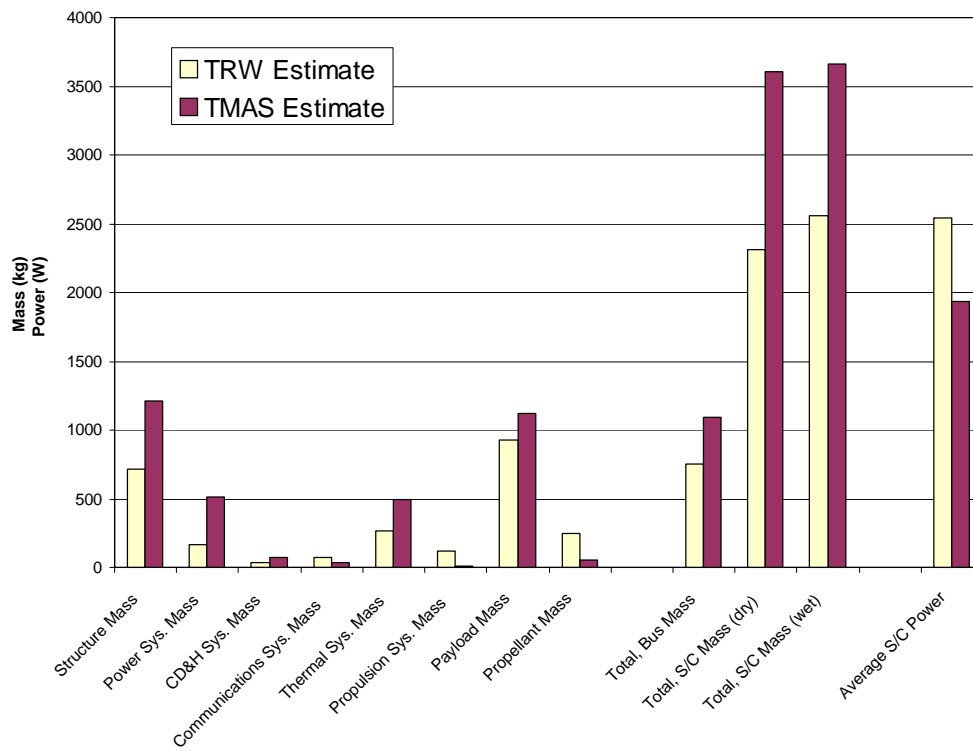
where the limits of integration denote time steps of one day,  $i$  is an index indicating a particular operational state,  $n$  is the total number of operational states ( $n=5$  for the eight-collector aperture TPF example),  $I_s$  is the survey mode imaging rate,  $I_m$  is the medium spectroscopy mode imaging rate,  $I_d$  is the deep spectroscopy mode imaging rate, and  $P_i(t)$  is the probability of being in any state  $i$  as a function of time  $t$  from Eqn. 4.13. The imaging mode determines the SNR needed to obtain the desired information on the target. This required SNR in turn determines the required integration time, and thus the imaging rate in a particular mode. For example, the required SNR for a survey, in which the purpose is to determine whether or not a planet exists in the habitable zone around the target star, is only five. The medium spectroscopy imaging mode searches for spectral lines of carbon dioxide and water indicative of a planetary atmosphere and requires an SNR of ten. Deep spectroscopy searches for spectral lines of ozone and methane that might signify the presence of a habitable environment or even the existence of life itself, and requires an SNR of 25. Accordingly, surveys require the shortest integration time, and thus the survey mode imaging rate  $I_s$  is greater than  $I_m$ , and  $I_m$  is greater than  $I_d$ . The limits of integration in Eqn. 4.15 are based on a sample five year TPF mission profile [Beichman et al, 1999].

The total system performance is obtained by summing fifteen separate utility functions, where each row in Eqn. 4.15 denotes a separate year in the five-year mission and each column denotes a different mission mode.

### **Benchmarking the TPF GINA Model**

The TPF GINA model has been benchmarked [Curtis et al, 1999] against point designs proposed by Ball [Noecker et al, 1997], TRW [Wehner et al, 1997], and Lockheed Martin [Francis et al, 1997]. The results of evaluating each of these design configurations using the GINA model were compared to the TPF system parameters generated by the contractors. When discrepancies were found, they were addressed by first examining the assumptions behind the model, and then by modifying the required module(s) to increase model fidelity. The goal of this verification process was to gain confidence in the results produced by the TPF GINA model, not to make the output parameters exactly match those from the other designs.

Overall, the test cases validated the TPF GINA model. The parameter comparisons primarily involved subsystem mass and power estimates. Figure 4.19 shows the validation data for the comparison between TPF GINA model (TMAS) data and the structurally connected interferometer conceptual design developed by TRW [Wehner et al, 1997]. In this case, the significant discrepancies between the two estimates concern the spacecraft structure and the power and propulsion subsystems. The primary reason for these differences is that the TRW design minimized mass using an ultra lightweight truss and lightweight solar concentrators. The GINA model accounts for certain technological advances, but uses more conservative assumptions than those made by TRW. Additionally, the propulsion subsystem in the TRW design includes the orbit transfer propellant, while the GINA model estimate does not. Instead, it assumes direct orbital insertion by the launch vehicle. For a complete discussion of all the benchmark cases, please see the reference by Curtis et al [Curtis et al, 1999].



**Figure 4.19** Sample TPF GINA Model Benchmark with a TRW Point Design [Curtis et al, 1999]

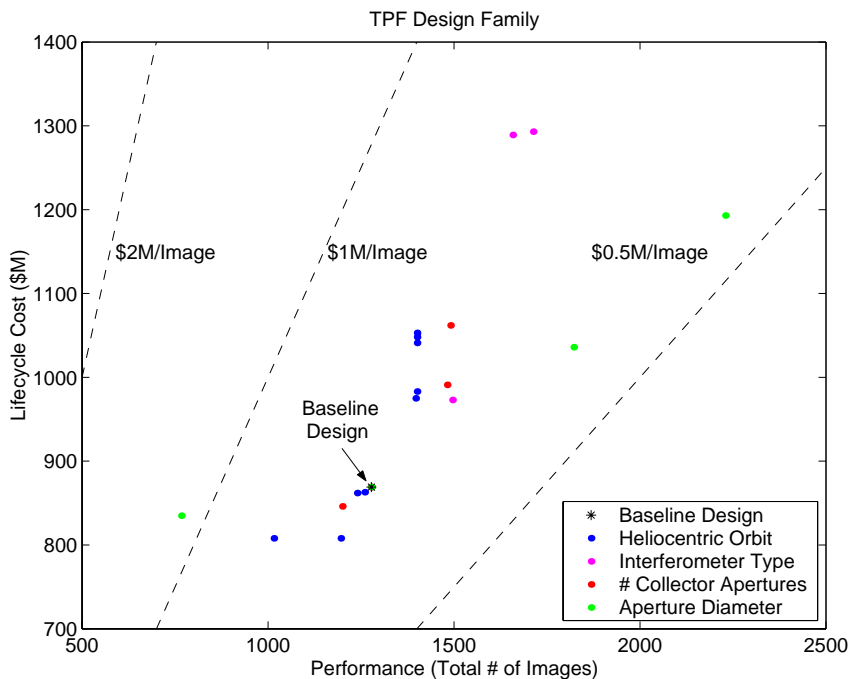
## 4.2.6 GINA Step 6 – Explore the System Trade Space

With the GINA model now complete, the TPF trade space may be explored. This will be done in step two of the MMDOSA methodology.

## 4.3 MMDOSA Step 2 – Perform Univariate Studies

Univariate studies may be used to find the general shape of a "design family" within the system trade space. A design family is created by establishing a baseline design and then varying each parameter within the design vector across its entire range while holding all other components of the design vector at their baseline values. Table 4.3 lists the baseline design vector for the design family illustrated in Figure 4.20.

Figure 4.20 illustrates a sample TPF design family, with the total number of images on the x-axis and lifecycle cost on the y-axis. Each dot represents a different TPF system archi-



**Figure 4.20** Example TPF Design Family

**TABLE 4.3** Baseline TPF Design Vector

Design Vector	Value
Heliocentric Orbital Radius	3.0 AU
Collector Connectivity/Geometry	SCI-1D
# Collector Apertures	6
Diameter of Collector Apertures	2 m

texture within the same design family. The black asterisk represents the baseline design (Table 4.3) for this particular family, and the diagonal lines are iso-CPF lines (i.e. lines of constant cost per image). Notice that the heliocentric orbital radius, interferometer type, and number of collector apertures design vector parameters all increase lifecycle cost faster than they increase lifecycle performance. Conversely, varying the aperture diameter, denoted by the green dots, induces the largest spread in the system performance. Thus, it appears that, at least in this local trade space, aperture diameter exerts the greatest influence on system performance and the cost per image metric. This issue will be revisited within the context of the global trade space in Section 4.6. Figure 4.23, Figure 4.21,

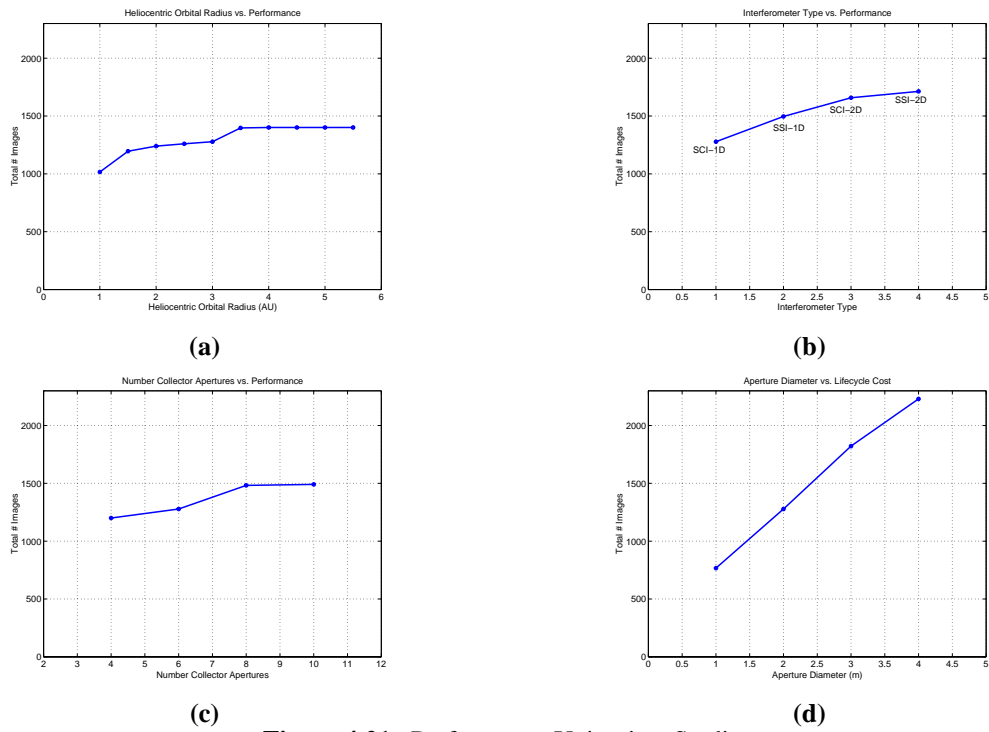


Figure 4.21 Performance Univariate Studies

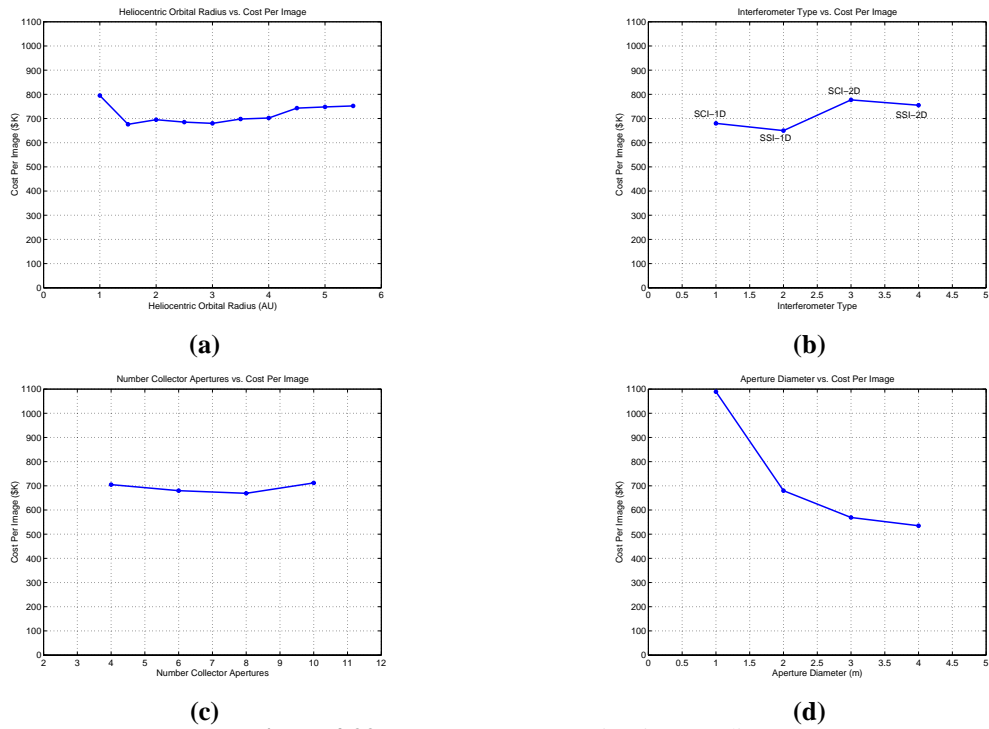
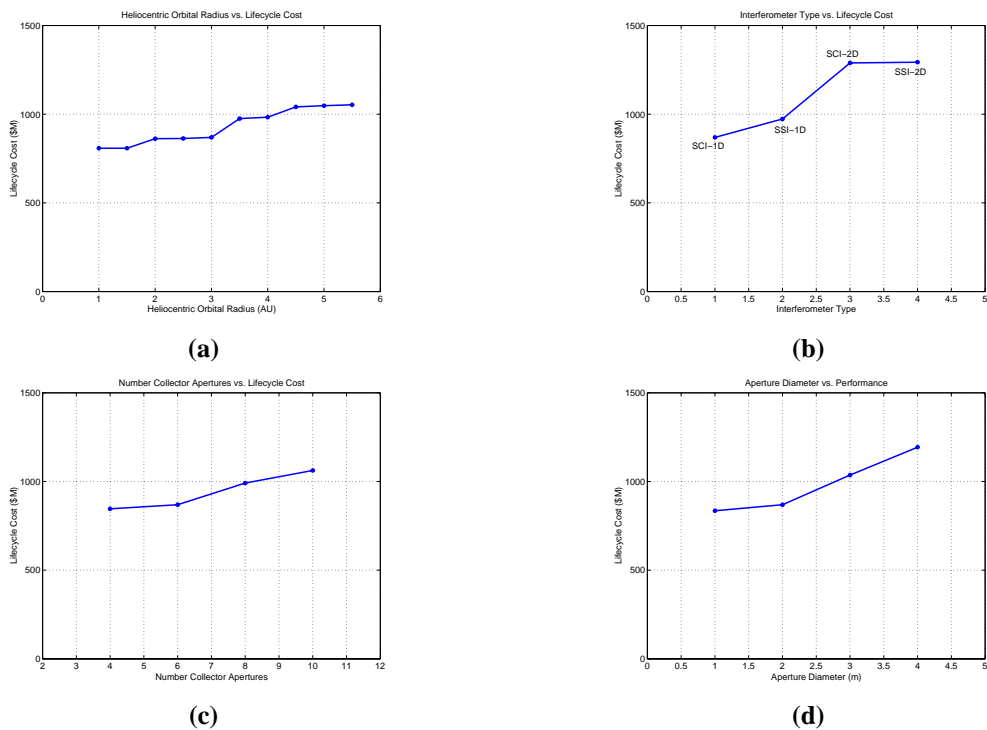


Figure 4.22 Cost Per Image Univariate Studies



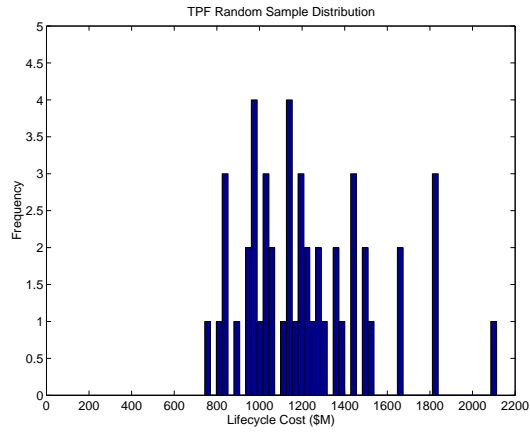
**Figure 4.23** Lifecycle Cost Univariate Studies

and Figure 4.22 illustrate univariate studies executed on the TPF GINA model to verify model fidelity and gain an initial feel for the trade space. For the complete set of univariate studies on the TPF GINA model, please see the reference by Curtis et al [Curtis et al, 1999].

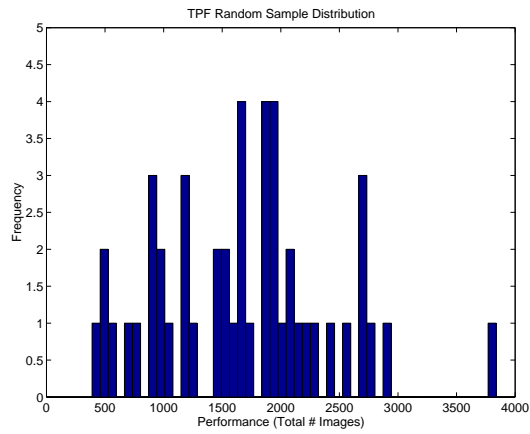
### 4.4 MMDOSA Step 3 – Take a Random Sample of the Trade Space

Within the TPF global trade space of 640 architectures, 48 architectures – 7.5% of the trade space – were randomly sampled. Figure 4.24 illustrates the results from the random sample for each metric which the TPF design is simultaneously optimized with respect to in Section 4.5.2. Table 4.4 summarizes the statistics from this random sample. Please see Section 4.5.2 for details on how these statistics are used to tailor the MDO search algorithms.

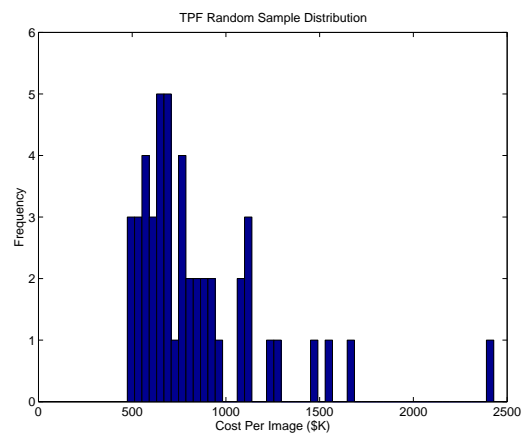




(a)



(b)



(c)

Figure 4.24 Random Sample Results for Lifecycle Cost (a), Performance (b), and Cost Per Image (c)

**TABLE 4.4** Statistics from a Random Sample of the TPF Global Trade Space

<b>Parameter</b>	<b>Lifecycle Cost (\$B)</b>	<b>Performance (# Images)</b>	<b>Cost Per Image (\$K)</b>
Maximum	2.11	3839	2429
Minimum	0.74	390	474
Range	1.37	3449	1955
Mean	1.23	1681	849
Median	1.18	1659	734
Standard Deviation	0.30	731	367

These statistics can be used to compute the 95% confidence interval (95%CI) for each parameter  $i$  in the objective function:

$$95\% \text{ CI} = \bar{x}_i \pm \frac{2v(x_i)}{\sqrt{n}} \quad (4.16)$$

where  $\bar{x}_i$  is the sample mean value for parameter  $i$ ,  $v(x_i)$  is the sample standard deviation for parameter  $i$ , and  $n$  is the size of the random sample. Table 4.5 lists the 95%CI of the mean value for the different architecture attributes that were measured in the random sample. This information tells the systems engineer what the likely characteristics of an "average" design will be. For TPF, it can be said with a 95% confidence that a randomly selected set of TPF system architectures will exhibit on average a lifecycle cost of approximately \$1.23B, a performance of 1681 images, and a cost per image of \$849K.

**TABLE 4.5** TPF 95% Confidence Intervals for the Global Trade Space Mean

<b>Parameter</b>	<b>95%CI</b>
Lifecycle Cost	\$1.14B - \$1.31B
Performance	1470 images - 1892 images
Cost Per Image	\$743K - \$955K

The systems engineer is not interested in creating an "average" design for the TPF system, however. Rather, the systems engineer wishes to find the best system architecture(s) based on the metric(s) of interested to the customer. Thus, the TPF architectures found via the MDO algorithms in step four of the MMDOSA methodology should rate much better

than the 95%CI parameter values listed in Table 4.5. In fact, the data gathered in the random sample of the trade space (Table 4.4) can be used to place initial bounds on each of the design parameters to be optimized. For example, the minimum CPF TPF architectures found via MDO algorithms should yield a lower CPF than the minimum CPF value of \$474K in Table 4.4, which was the lowest CPF value measured in the random sample.

## 4.5 MMDOSA Step 4 – Formulate and Apply MDO Algorithm(s)

Two separate MDO approaches have been applied to the exploration of the TPF trade space. These approaches are summarized in the test matrix in Table 4.6. In the first test case, a single objective simulated annealing algorithm is used to search the TPF global trade space for architectures that minimize the cost per function metric. For the TPF mission, the CPF metric is defined as the cost per image (CPI). In the second test case, the design scope is expanded from that of a single objective optimization problem to a more difficult multiobjective optimization problem in which the systems engineer simultaneously attempts to minimize lifecycle cost and maximize system performance.

**TABLE 4.6** MDO Test Matrix for TPF Case Study

<b>Test Case</b>	<b>Structure</b>
<b>Approach 1</b>	
Type	Single Objective Optimization
Objective Function	Minimize CPF (i.e. Cost Per Image)
Trade Space Description	TPF Global Trade Space
Trade Space Size	640 system architectures
<b>Approach 2</b>	
Type	Multiobjective Optimization
Objective Function	Minimize Lifecycle Cost and Maximize System Performance
Trade Space Description	TPF Global Trade Space
Trade Space Size	640 system architectures

### 4.5.1 Approach 1 – Single Objective Optimization

#### Objective

In the first stage of the test matrix, the objective is to find the most cost-effective TPF system architectures without completely enumerating the trade space. This is done by applying the single objective simulated annealing algorithm described in Section 3.5.1. The sole objective is to minimize the CPF metric, and the algorithm is terminated after 48 iterations. Thus, this first design problem may be characterized as a single objective DSS architecture conceptual design problem.

#### Problem Formulation

The first TPF architecture design optimization problem may be formulated mathematically as:

$$\begin{array}{ll}
 \text{Objective:} & \text{Min} \left( \frac{\sum_{y=1}^5 \Phi_y(\Gamma)}{\sum_{y=1}^5 \Psi_y(\Gamma)} \right) \\
 \text{Constraints:} & \text{Subject to} \\
 \text{Isolation} & \theta_r \geq 20 \text{ milliarc sec} \\
 & \Omega \leq 10^{-6} \\
 \text{Integrity} & \text{SNR}_s \geq 5 \\
 & \text{SNR}_{ms} \geq 10 \\
 & \text{SNR}_{ds} \geq 25 \\
 \text{Availability} & A_v \geq 0.95 \\
 \text{Bounds} & 1.0 \text{ AU} \leq \gamma_1 \leq 5.5 \text{ AU} \\
 & 4 \leq \gamma_3 \leq 10 \\
 & 1 \text{ m} \leq \gamma_4 \leq 4 \text{ m}
 \end{array} \tag{4.17}$$

where  $y$  is the year of the mission,  $\Phi$  is cost,  $\Psi$  is the number of images (i.e. surveys + medium spectroscopies + deep spectroscopies),  $\theta_r$  is angular resolution,  $\Omega$  is null depth,  $\text{SNR}_s$  is the survey mode signal-to-noise ratio (i.e. photon count received from the planet over the photon count received from all noise sources),  $\text{SNR}_{ms}$  is the medium spectroscopy mode signal-to-noise ratio,  $\text{SNR}_{ds}$  is the deep spectroscopy mode signal-to-noise ratio, and  $A_p$  is the system availability. To evaluate a TPF architecture on the basis of the objective function and constraints in Eqn. 4.17, the design vector is entered into the TPF GINA model. Table 4.2 lists the design vector variables along with the allowable range of values for each of these variables.

### Algorithm Design

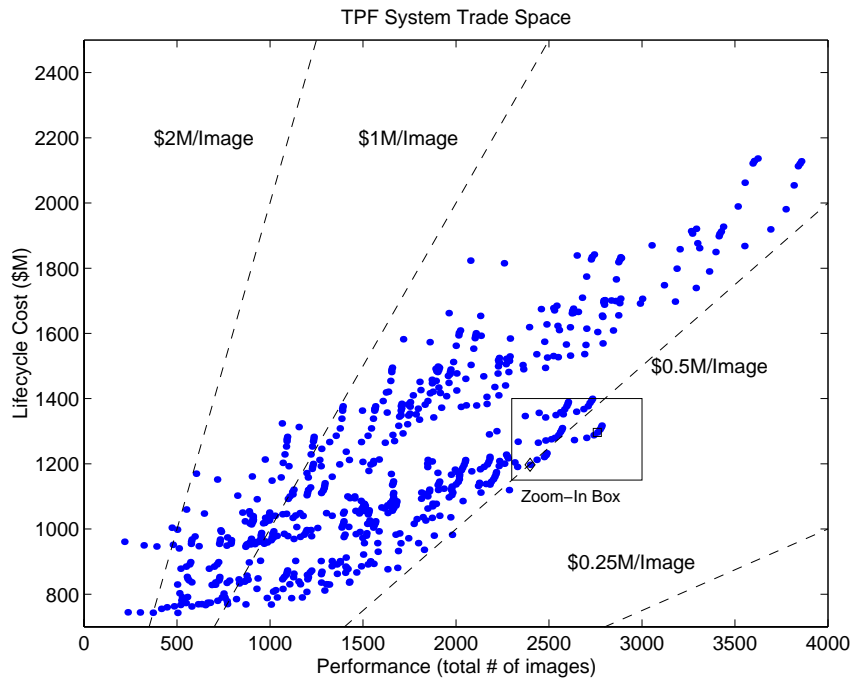
A 2-DOF single objective simulated annealing algorithm has been applied to the TPF conceptual design optimization problem in Eqn. 4.17. Table 4.7 lists the cooling schedule used by the single objective simulated annealing algorithm. Initial system temperature was set to 1000 and was reduced by a factor of 0.85 after each iteration. Each trial of the algorithm was terminated after 48 iterations.

**TABLE 4.7** Simulated Annealing Cooling Schedule

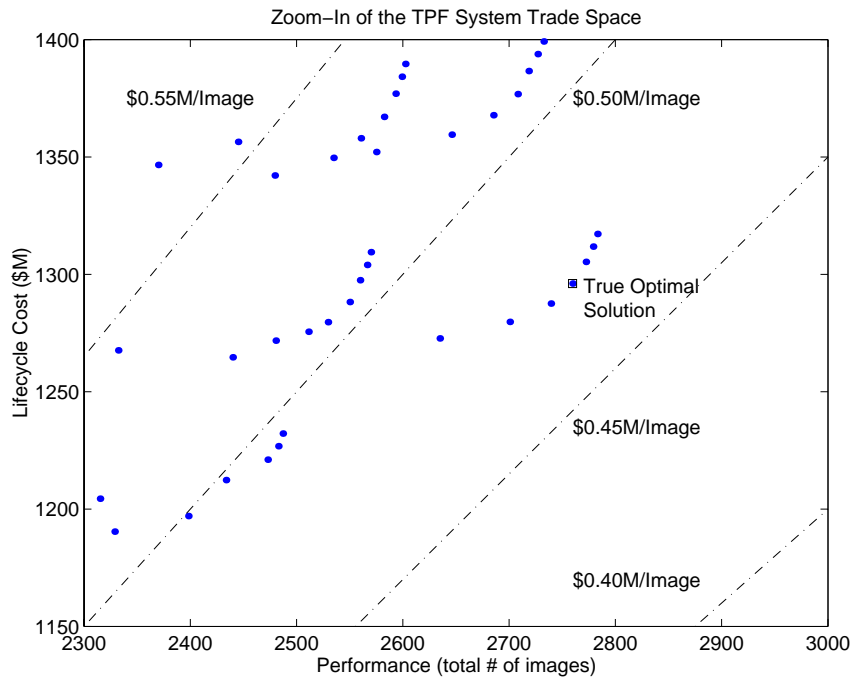
Cooling Schedule Parameters	Value
Initial System Temperature	1000 \$K/Image
Temperature Reduction Factor	0.85
Number Iterations	48

### Results

The plot in Figure 4.25a illustrates the global TPF trade space for the conceptual design case study as enumerated in Section 3.5.1. The dashed diagonal lines are iso-CPI lines. All the system architectures that fall along these lines exhibit the same cost-effectiveness and are identical from the perspective of the objective function in Eqn. 4.17, albeit with different levels of performance and lifecycle cost. Figure 4.25b zooms in on the region of most cost-effective architectures within the trade space. The single objective simulated



(a)



(b)

**Figure 4.25** TPF Global Trade Space (a) and Trade Space Zoom-In On the Region of Most Cost-Effective Architectures

annealing algorithm successfully found these architectures after searching only 7.5% of the global trade space. Table 4.8 lists the ten most cost-effective TPF architectures identified by the MDO algorithm.

**TABLE 4.8** Most Cost-Effective TPF Architectures Found by the Simulated Annealing Algorithm

Parameter	1	2	3	4	5	6	7	8	9	10
Heliocentric Orbital Radius (AU)	4.0	3.5	4.5	5.0	5.5	3.0	2.5	1.5	4.5	5.0
Collector Connectivity/Geometry*	SCI-2D	SCI-2D	SCI-2D	SCI-2D	SCI-2D	SCI-2D	SCI-2D	SCI-2D	SCI-2D	SCI-2D
# Collector Apertures	8	8	8	8	8	8	8	8	6	6
Aperture Diameter (m)	4	4	4	4	4	4	4	4	4	4
Lifecycle Cost (\$B)	1.30	1.29	1.31	1.31	1.32	1.28	1.27	1.12	1.22	1.23
Performance (# Images)	2760	2739	2772	2779	2783	2701	2635	2285	2473	2483
CPF (\$K/image)	469.6	470.0	470.8	472.0	473.3	473.9	483.0	490.0	493.8	494.1

## Summary

In this section, the TPF architecture conceptual design problem has been modeled as a single objective optimization problem, and a simulated annealing algorithm has been applied to search the global trade space for the most cost-effective architectures without completely enumerating the trade space. Several cost-effective architectures were found, including the true optimal solution identified in Section 3.5.1.

## 4.5.2 Approach 2 – Multiobjective Optimization

### Objective

In the second stage of the test matrix, the TPF system architecture conceptual design problem is modeled as a multiobjective optimization problem with the intention of finding the Pareto optimal set of architectures without completely enumerating the trade space. This is done by applying the multiobjective multiple solution simulated annealing algorithm developed in Section 3.5.2, which attempts to satisfy simultaneously the competing objectives of minimizing lifecycle cost and maximizing system performance (i.e. maximizing the total number of images collected). The algorithm is terminated after 48 iterations. This second conceptual design problem may be characterized as a multiobjective DSS architecture design problem.

### Problem Formulation

This multiobjective conceptual design optimization problem may be formulated mathematically as:

$$\begin{aligned}
 \text{Objective:} & \quad \text{Min} \left( \sum_{y=1}^5 \Phi_y(\Gamma) \right) \text{ AND } \text{Max} \left( \sum_{y=1}^5 \Psi_y(\Gamma) \right) \\
 \text{Constraints:} & \quad \textit{Subject to} \\
 \text{Isolation} & \quad \theta_r \geq 20 \text{ milliarc sec} \\
 & \quad \Omega \leq 10^{-6} \\
 \text{Integrity} & \quad \text{SNR}_s \geq 5 \\
 & \quad \text{SNR}_{ms} \geq 10 \\
 & \quad \text{SNR}_{ds} \geq 25 \\
 \text{Availability} & \quad A_v \geq 0.95 \\
 \text{Bounds} & \quad 1.0 \text{ AU} \leq \gamma_1 \leq 5.5 \text{ AU} \\
 & \quad 4 \leq \gamma_3 \leq 10 \\
 & \quad 1 \text{ m} \leq \gamma_4 \leq 4 \text{ m}
 \end{aligned} \tag{4.18}$$

To evaluate a TPF system architecture on the basis of the objective function and constraints in Eqn. 4.18, the design vector is entered into the TPF GINA model. Table 4.2 lists the design vector variables along with the allowable range of values for each of these variables.

In addition to the change in the mathematical formulation, the multiobjective optimization problem also requires a change in the internal mechanics of the simulated annealing algorithm. As explained in Section 3.5,  $\Delta$  is a parameter that is used to determine the probability with which the algorithm may move to a "worse" solution with the intent of escaping local minima within the nonconvex trade space. In the single objective case,  $\Delta$  is simply the difference in the objective function value between successive steps of the algorithm.



For multiobjective problems, however,  $\Delta$  must take into account the difference in successive steps between all objective functions – that is for all  $N$  dimensions of the problem.

The formula for computing the  $\Delta$  parameter for a multiobjective optimization problem with  $N$  dimensions in the objective function (i.e.  $N$  design goals or  $N$  decision criteria) is:

$$\Delta = \frac{\sum_{n=1}^N \frac{E_n(\Gamma_{i+1}) - E_n(\Gamma_i)}{\delta_n}}{N} \tag{4.19}$$

where  $i$  is the iteration number within the simulated annealing algorithm, and  $\delta_n$  is the maximum range in dimension  $n$  observed in the random sample. Depending upon the distribution of the random sample,  $\delta_n$  may be replaced by the  $1\sigma_n$  range,  $2\sigma_n$  range, or  $3\sigma_n$  range for each dimension  $n$  in the objective function.

Table 4.4 lists the results for each metric in the objective function in Eqn. 4.18 from the random sample of the TPF trade space. Using this information, the  $\Delta$  parameter for the multiobjective optimization problem in Eqn. 4.18 may be computed as:

$$\Delta = \frac{\frac{\Phi(\Gamma_{i+1}) - \Phi(\Gamma_i)}{\$1.37 \times 10^9} + \frac{\Psi(\Gamma_i) - \Psi(\Gamma_{i+1})}{3449}}{2} \tag{4.20}$$

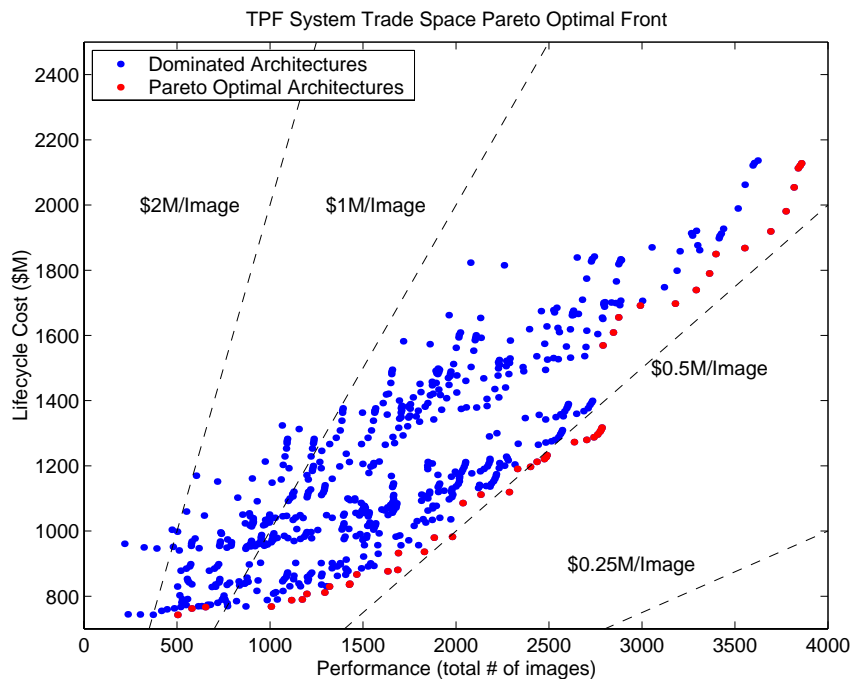
Table 4.9 lists the cooling schedule used by the single objective simulated annealing algorithm. The initial system temperature was set to one and was reduced by a factor of 0.85 after each iteration. Each trial of the algorithm was terminated after 48 iterations.

**TABLE 4.9** Simulated Annealing Cooling Schedule

Cooling Schedule Parameters	Value
Initial System Temperature	1
Temperature Reduction Factor	0.85
Number Iterations	48

## Results

Figure 4.26 illustrates the Pareto optimal set of TPF architectures within the global trade space for the conceptual design case study as enumerated in Section 3.5.1. The multiobjective multiple solution simulated annealing algorithm successfully found these Pareto optimal architectures after searching only 7.5% of the global trade space. Figure 4.27 lists the Pareto optimal TPF architectures identified by the MDO algorithm. Within this Pareto optimal set, the systems engineer cannot improve the performance of the TPF architecture without also increasing lifecycle cost. Notice that several families of architectures inhabit the Pareto boundary, with structurally connected architectures occupying the low through medium cost and performance range of the boundary, and separated spacecraft architectures occupying the high cost, high performance end of the boundary.



**Figure 4.26** TPF Trade Space Global Pareto Boundary

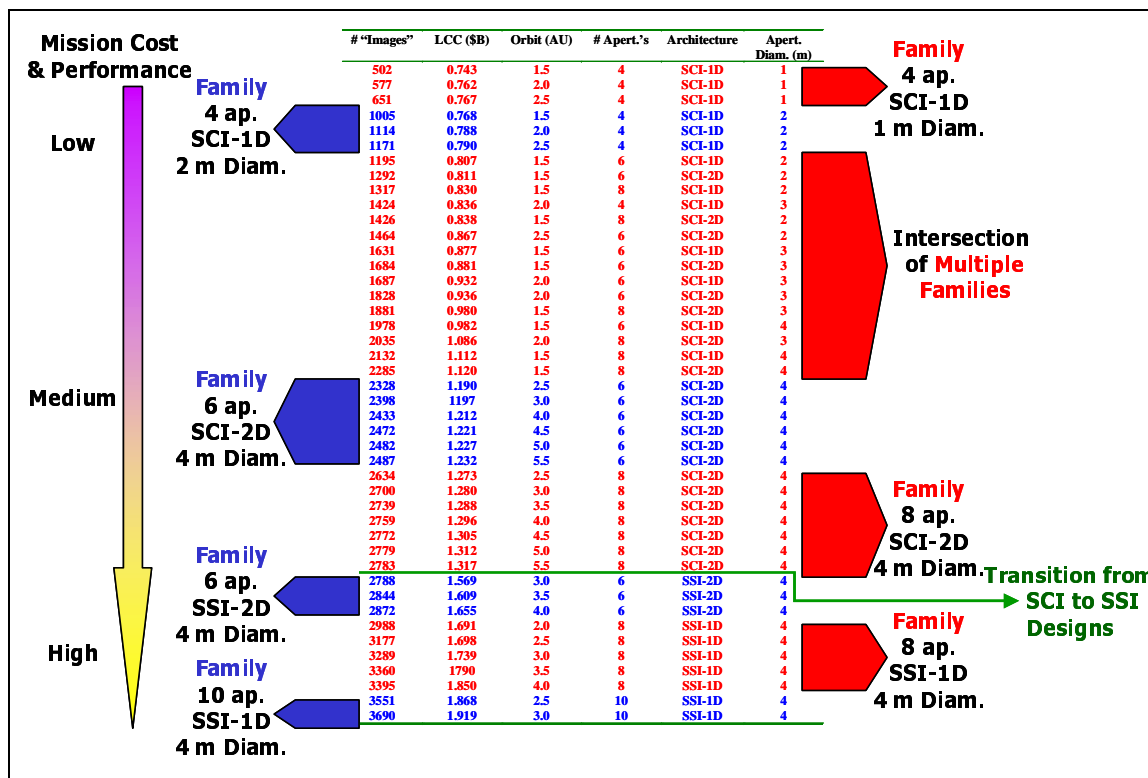


Figure 4.27 Pareto Optimal Architectures Found by the MDO Algorithm

Summary

In this section, the TPF architecture conceptual design problem has been modeled as a multiobjective optimization problem, and the multiobjective multiple solution simulated annealing algorithm has been applied to search the TPF global trade space for the Pareto optimal set of system architectures without completely enumerating the trade space. Several families of Pareto optimal architectures were found ranging from low cost, low performance designs to high cost, high performance designs.

4.6 MMDOSA Step 5 – Interpret Results (Sensitivity Analysis)

Figure 4.28 illustrates the ANOVA results for the TPF trade space. As one can see, aperture diameter exerts by far the greatest relative influence on the cost per image metric. In comparison, the heliocentric orbital radius, interferometer type, and number of apertures exert much less influence on the cost per image metric. ANOVA provides the systems

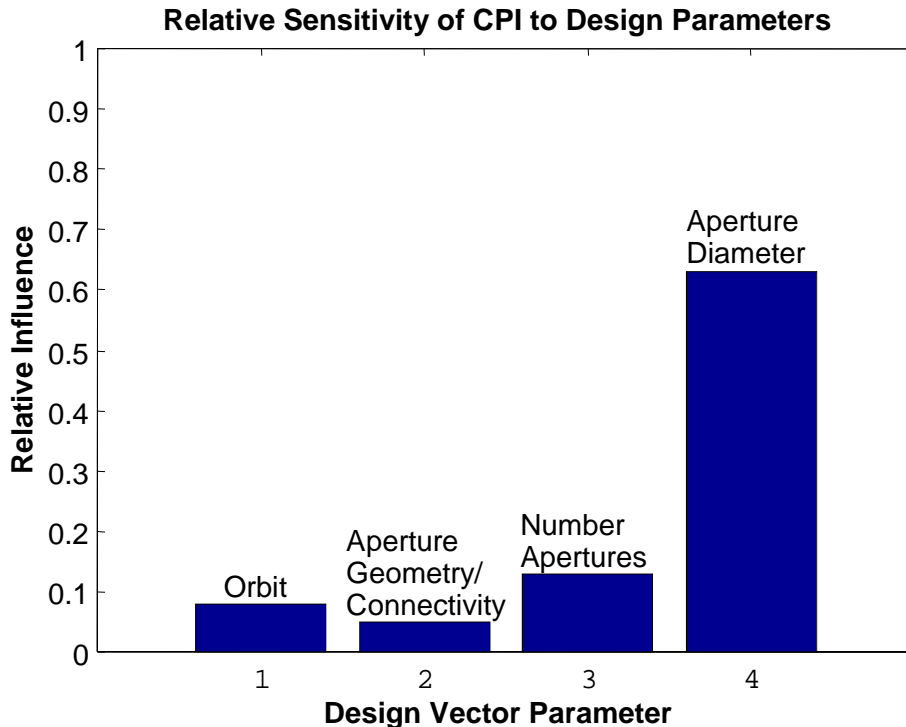
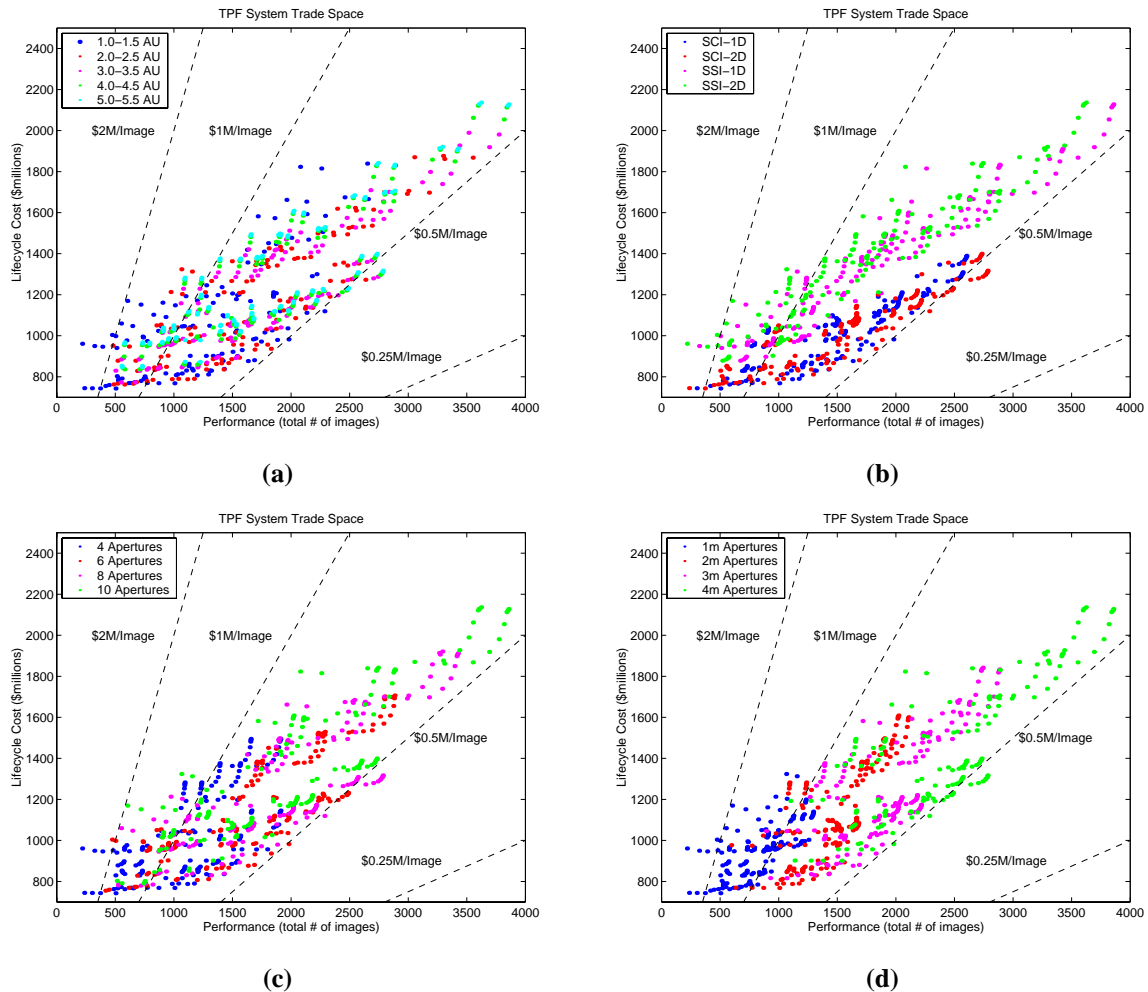


Figure 4.28 ANOVA Results for the TPF Trade Space

engineer with information on which design parameters give the most leverage over various aspects of the design. The ANOVA results may also be used to guide technology investment decisions. For example, if the TPF program has a limited technology investment budget and wishes to improve the cost-effectiveness of the mission, then it appears from the ANOVA results that those funds would be best spent on technologies which will yield light weight and low cost large diameter collector apertures.

Further insight into the TPF trade space may be gained by viewing the trade space from different perspectives. Figure 4.29 illustrates four different views of the TPF trade space. In Figure 4.29a, the design architectures are color-coded according to their heliocentric orbital radius. While the maximum cost and highest performance does scale with the orbital radius, the difference in these values from 1.0 AU to 5.5 AU is less than 1000 images and \$400M respectively. Figure 4.29b color-codes the architectures according to interferometer type. Notice two distinct bands differentiating the structurally connected



**Figure 4.29** Different Perspectives of the TPF Trade Space as Viewed According to Heliocentric Orbital Radius (a), Collector Connectivity/Geometry (b), Number of Collector Apertures (c), and Aperture Diameter (d)

and separated spacecraft architectures. The structurally connected architectures are more cost effective than the separated spacecraft architectures according to the cost per function metric. However, the best achievable performance of structurally connected architectures is limited to approximately 2800 images over five years, while separated architectures may collect almost 3900 images during the same time period. This plot also reveals that two-dimensional arrays yield slightly better performance for the equivalent cost. In Figure 4.29c, the design architectures are color-coded according to the number of apertures in the interferometer. Here there is a distinct trend between the number of apertures, maximum performance, and maximum lifecycle cost. The difference in these values

between four apertures and ten apertures is approximately 2000 images and \$700M. Finally, Figure 4.29d, color-coded according to aperture diameter, illustrates an even stronger correlation between aperture diameter, maximum performance, and maximum lifecycle cost; with the difference in these values between one meter apertures and four meter apertures approximating 2700 images and \$900M respectively. These observations corroborate the ANOVA results in Figure 4.28.

## 4.7 MMDOSA Step 6 – Iterate

Because the TPF trade space was completely enumerated to assess the effectiveness of the different MDO algorithms tested in Section 3.5 and the single objective simulated annealing algorithm was able to find the optimal family of design solutions after a single trial, it is not necessary to iterate in this case study. Rather, multiple trials from different randomly selected starting points may be executed.

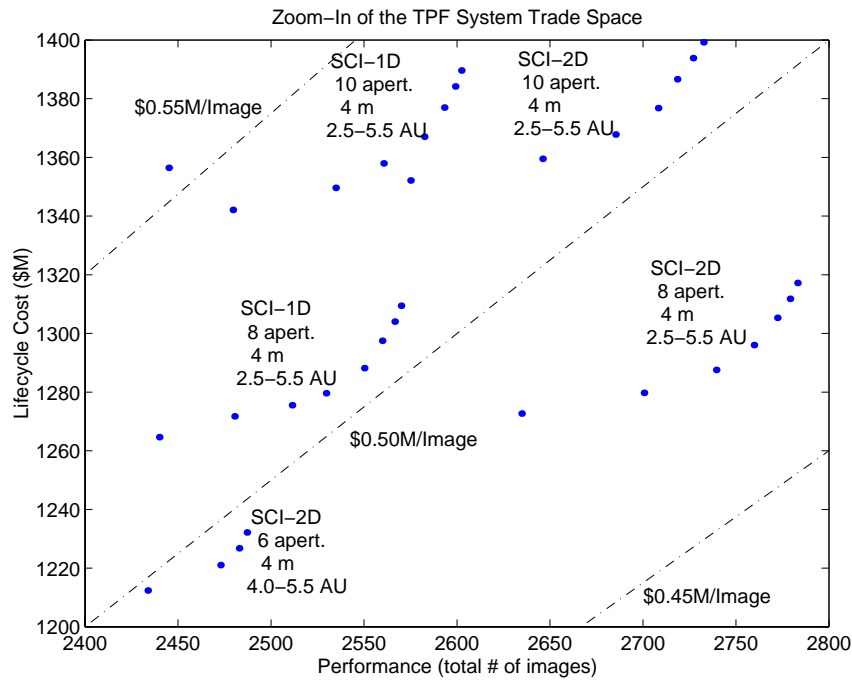
## 4.8 MMDOSA Step 7 – Converge on Best System Architectures

Having applied the first six steps of the MMDOSA methodology to the TPF conceptual design problem, the best architectures for the NASA Origins program to carry forward into the next program phase, Phase B, may be recommended. Architectures are recommended for each of the following design scenarios:

1. The NASA Origins program wishes to select the most cost-effective TPF system architecture as measured by the cost per image metric.
2. The NASA origins program wishes to select a TPF architecture that simultaneously minimizes lifecycle cost and maximizes the total number of images collected by the system.

### 4.8.1 Recommended Most Cost-Effective Architectures

Figure 4.30 zooms in on the region of most cost-effective architectures within the TPF global trade space according to the cost per image metric. Each arc in the figure



**Figure 4.30** Most Cost-Effective TPF Design Families

represents a different TPF design family. These optimal families are listed in Table 4.10 in order of decreasing cost-effectiveness.

**TABLE 4.10** Recommended Families of Architectures for Phase B

Design Vector	Family 1	Family 2	Family 3	Family 4	Family 5
Heliocentric Orbital Radius	2.5-5.5 AU	4.0-5.5 AU	2.5-5.5 AU	2.5-5.5 AU	2.5-5.5 AU
Collector Connectivity/Geometry	SCI-2D	SCI-2D	SCI-1D	SCI-2D	SCI-1D
# Collector Apertures	8	6	8	10	10
Aperture Diameter	4 m	4 m	4 m	4 m	4 m

Closer inspection of the recommended architectures reveals distinct patterns. All five families of architectures use four meter diameter apertures and a structurally connected interferometer. These families span a wide range of orbits beyond 2.5 AU. It appears that within 2.5 AU, the local zodiacal intensity significantly reduces the performance of the interferometer and yields it no longer cost-effective. While separated spacecraft interferometers can produce more images (Figure 4.29b), they are not as cost-effective as struc-

turally connected interferometers. These results point to total collector aperture area as the single most important parameter in developing a cost-effective TPF architecture. This observation corresponds with the observations from the sensitivity analysis (Figure 4.28).

## 4.8.2 Recommended Pareto Optimal Architectures

Figure 4.27 listed the complete set of Pareto optimal architectures within the global trade space. This set of architectures defines a boundary along which the systems engineer cannot improve system performance without also increasing lifecycle cost. Table 4.11 lists the best Pareto optimal TPF architecture as a function of the program budget. As the budget increases, greater levels of performance may be achieved. These are the architectures which should be carried forward into Phase B of the TPF program if NASA wishes to select an architecture that simultaneously minimizes lifecycle cost and maximizes the number of images collected.

**TABLE 4.11** Best Pareto Optimal TPF Architectures Ordered as a Function of Program Budget

<b>Parameter</b>	<b>\$0.75B Budget</b>	<b>\$1.00B Budget</b>	<b>\$1.25B Budget</b>	<b>\$1.50B Budget</b>	<b>\$1.75B Budget</b>	<b>\$2.00B Budget</b>
Heliocentric Orbital Radius (AU)	1.5	1.5	5.5	5.5	3.0	3.0
Collector Connectivity/Geometry	SCI-1D	SCI-1D	SCI-2D	SCI-2D	SSI-1D	SSI-1D
# Collector Apertures	4	6	6	8	8	10
Aperture Diameter (m)	1	4	4	4	4	4
Lifecycle Cost (\$B)	0.74	0.98	1.23	1.32	1.74	1.92
Performance (# Images)	502	1978	2487	2783	3289	3690
CPF (\$K/Image)	1480	496	495	473	529	520

## 4.9 Conclusions

This chapter has illustrated the application of the MMDOSA methodology to the conceptual design of the NASA Origins Terrestrial Planet Finder mission. First, the GINA methodology was applied to create a GINA simulation model for the system. Next, univariate studies and random sampling were executed to obtain initial information on the trade space. With this information in hand, single objective and multiobjective simulated



annealing algorithms were applied to search for the most cost-effective architectures and the Pareto optimal architectures within the trade space. After a sensitivity analysis, specific architectures were identified and recommended for further detailed study in the next phase of the TPF program.

Figure 4.31 illustrates the complete TPF trade space, including the Pareto optimal set of architectures; the most cost-effective minimum cost per image family of architectures; and various point design architectures developed by NASA JPL, Ball Aerospace, Lockheed Martin, and TRW. Table 4.12 through Table 4.16 compares the lifecycle cost, mission performance, and system cost per function as computed by the GINA model for various NASA and industry point design architectures with the following three best TPF architectures found by the MMDOSA methodology:

1. The most cost-effective architecture (i.e. minimizes the cost per function metric).

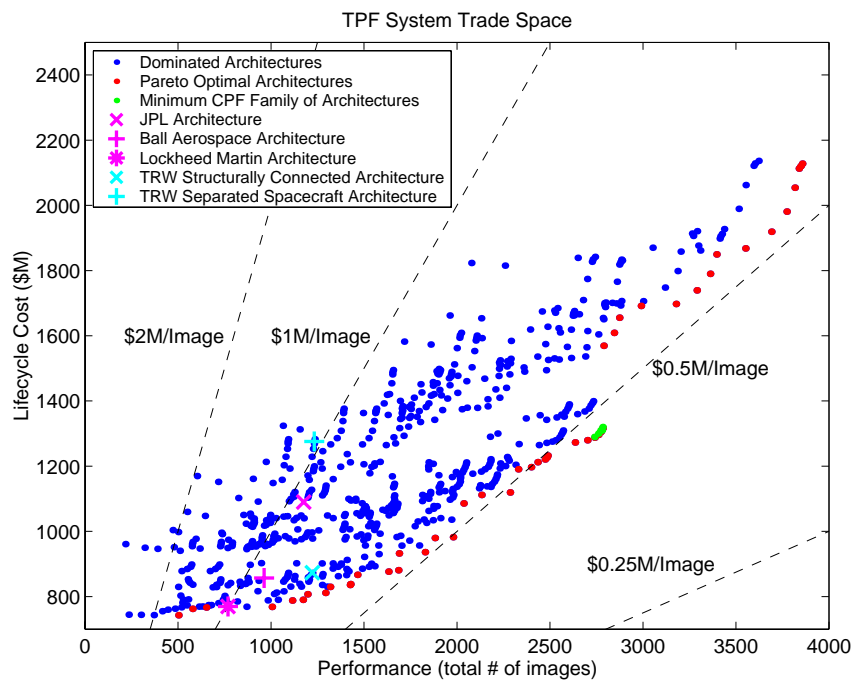


Figure 4.31 TPF System Trade Space

2. The Pareto optimal architecture that provides equivalent performance at a reduced lifecycle cost.
3. The Pareto optimal architecture that provides improved performance for an equivalent lifecycle cost.

Table 4.12 compares the NASA JPL reference mission point design with the best designs found by the MMDOSA methodology. The JPL architecture places a four 3.5 m aperture separated spacecraft one-dimensional interferometer in a 1.0 AU heliocentric orbit [Beichman et al, 1999]. The minimum CPF architecture reduces the cost per image of the system by 49%. The Pareto optimal equivalent performance architecture reduces lifecycle cost by 28%, and the Pareto optimal equivalent lifecycle cost architecture improves system performance by 73%. In each instance, the MMDOSA methodology identified better, more cost-effective TPF architectures than the NASA JPL reference mission point design.

**TABLE 4.12** Comparison of NASA JPL Point Design with Designs Found by the MMDOSA Methodology

Architecture	JPL Reference Design	MMDOSA Minimum CPF Architecture	Percent Improvement	Pareto Optimal Equivalent Performance	Percent Improvement	Pareto Optimal Equivalent LCC	Percent Improvement
Orbit (AU)	1.0	4.0		2.5		2.0	
Conn./Geom.	SSI-1D	SCI-2D		SCI-1D		SCI-2D	
# Apertures	4	8		4		8	
Ap. Diameter (m)	3.5	4.0		2.0		3.0	
LCC (\$B)	1.09	1.30	-19%	0.79	28%	1.09	
# Images	1176	2760	134%	1171		2035	73%
CPF (\$K/Image)	927	470	49%	675	27%	536	42%

Table 4.13 compares the Ball Aerospace point design with the best designs found by the MMDOSA methodology. The Ball Aerospace architecture places a four 1.5 m aperture structurally connected one-dimensional interferometer in a 5.0 AU heliocentric orbit [Noecker et al, 1997]. The Ball Aerospace architecture actually uses a 0.5m-1.5m-1.5m-0.5m aperture diameter layout. However, in order to do a fair comparison with the results from MMDOSA, all the apertures in the Ball Aerospace architecture were set to 1.5 m. The minimum CPF architecture reduces the cost per image of the system by 47%. The

Pareto optimal equivalent performance architecture reduces lifecycle cost by 10%, and the Pareto optimal equivalent lifecycle cost architecture improves system performance by 52%. In each instance, the MMDOSA methodology identified better, more cost-effective TPF architectures than the Ball Aerospace point design.

**TABLE 4.13** Comparison of Ball Aerospace Point Design with Designs Found by the MMDOSA Methodology

Architecture	Ball Aerospace Design	MMDOSA Minimum CPF Architecture	Percent Improvement	Pareto Optimal Equivalent Performance	Percent Improvement	Pareto Optimal Equivalent LCC	Percent Improvement
Orbit (AU)	<b>5.0</b>	4.0		1.5		2.5	
Conn./Geom.	<b>SCI-1D</b>	SCI-2D		SCI-1D		SCI-2D	
# Apertures	<b>4</b>	8		4		6	
Ap. Diameter (m)	<b>1.5</b>	4.0		2.0		2.0	
LCC (\$B)	<b>0.86</b>	1.30	-51%	0.77	<b>10%</b>	0.87	
# Images	<b>963</b>	2760	187%	1005		1464	<b>52%</b>
CPF (\$K/Image)	<b>890</b>	470	<b>47%</b>	766	14%	594	33%

Table 4.14 compares the first TRW point design with the best designs found by the MMDOSA methodology. The first TRW architecture places a four 2.0 m aperture structurally connected one-dimensional interferometer in a 5.0 AU heliocentric orbit [Wehner et al, 1997]. The TRW architecture actually uses a 1.0m-2.0m-2.0m-1.0m aperture diameter layout. However, in order to do a fair comparison with the results from MMDOSA, all the apertures in the TRW architecture were set to 2.0 m. The minimum CPF architecture reduces the cost per image of the system by 34%. The Pareto optimal equivalent performance architecture reduces lifecycle cost by 7%, and the Pareto optimal equivalent lifecycle cost architecture improves system performance by 20%. In each instance, the MMDOSA methodology identified better, more cost-effective TPF architectures than the TRW point design.

Table 4.15 compares the second TRW point design with the best designs found by the MMDOSA methodology. The second TRW architecture places a four 2.0 m aperture separated spacecraft one-dimensional interferometer in a 5.0 AU heliocentric orbit [Wehner

**TABLE 4.14** Comparison of TRW SCI Point Design with Designs Found by the MMDOSA Methodology

Architecture	TRW SCI Design	MMDOSA Minimum CPF Architecture	Percent Improvement	Pareto Optimal Equivalent Performance	Percent Improvement	Pareto Optimal Equivalent LCC	Percent Improvement
Orbit (AU)	<b>5.0</b>	4.0		1.5		2.5	
Conn./Geom.	<b>SCI-1D</b>	SCI-2D		SCI-1D		SCI-2D	
# Apertures	<b>4</b>	8		6		6	
Ap. Diameter (m)	<b>2.0</b>	4.0		2.0		2.0	
LCC (\$B)	<b>0.87</b>	1.30	-49%	0.81	<b>7%</b>	0.87	
# Images	<b>1222</b>	2760	126%	1195		1464	<b>20%</b>
CPF (\$K/Image)	<b>714</b>	470	<b>34%</b>	678	5%	594	17%

et al, 1997]. The TRW architecture actually uses a 1.0m-2.0m-2.0m-1.0m aperture diameter layout. However, in order to do a fair comparison with the results from MMDOSA, all the apertures in the TRW architecture were set to 2.0 m. The minimum CPF architecture reduces the cost per image of the system by 55%. The Pareto optimal equivalent performance architecture reduces lifecycle cost by 37%, and the Pareto optimal equivalent lifecycle cost architecture improves system performance by 119%. In each instance, the MMDOSA methodology identified better, more cost-effective TPF architectures than the TRW point design.

**TABLE 4.15** Comparison of TRW SSI Point Design with Designs Found by the MMDOSA Methodology

Architecture	TRW SSI Design	MMDOSA Minimum CPF Architecture	Percent Improvement	Pareto Optimal Equivalent Performance	Percent Improvement	Pareto Optimal Equivalent LCC	Percent Improvement
Orbit (AU)	<b>5.0</b>	4.0		1.5		3.0	
Conn./Geom.	<b>SSI-1D</b>	SCI-2D		SCI-1D		SCI-2D	
# Apertures	<b>4</b>	8		6		8	
Ap. Diameter (m)	<b>2.0</b>	4.0		2.0		4.0	
LCC (\$B)	<b>1.28</b>	1.30	-2%	0.81	<b>37%</b>	1.28	
# Images	<b>1233</b>	2760	124%	1195		2700	<b>119%</b>
CPF (\$K/Image)	<b>1034</b>	470	<b>55%</b>	678	34%	474	54%

Table 4.16 compares the Lockheed Martin point design with the best designs found by the MMDOSA methodology. The Lockheed architecture places a four 2.0 m aperture structurally connected one-dimensional interferometer in a 1.0 AU heliocentric orbit [Francis

et al, 1997]. The Lockheed architecture actually uses a 1.0m-2.0m-2.0m-1.0m aperture diameter layout. However, in order to do a fair comparison with the results from MMDOSA, all the apertures in the Lockheed architecture were set to 2.0 m. The minimum CPF architecture reduces the cost per image of the system by 53%. The Pareto optimal equivalent performance architecture does not reduce lifecycle cost in this case, but the Pareto optimal equivalent lifecycle cost architecture does improve system performance by 31%.

**TABLE 4.16** Comparison of Lockheed Point Design with Designs Found by the MMDOSA Methodology

Architecture	Lockheed Design	MMDOSA Minimum CPF Architecture	Percent Improvement	Pareto Optimal Equivalent Performance	Percent Improvement	Pareto Optimal Equivalent LCC	Percent Improvement
Orbit (AU)	<b>1.0</b>	4.0		2.5		1.5	
Conn./Geom.	<b>SCI-1D</b>	SCI-2D		SCI-1D		SCI-1D	
# Apertures	<b>4</b>	8		4		4	
Ap. Diameter (m)	<b>2.0</b>	4.0		1.0		2.0	
LCC (\$B)	<b>0.77</b>	1.30	-69%	0.77	<b>0%</b>	0.77	
# Images	<b>769</b>	2760	259%	651		1005	<b>31%</b>
CPF (\$K/Image)	<b>1000</b>	470	<b>53%</b>	1183	-18%	766	23%

In all five test cases the MMDOSA methodology enabled the systems engineer to identify better, more cost-effective architectures than those previously considered by NASA and industry. In this manner, the MMDOSA methodology improves the quality of the work performed by systems engineers during the conceptual design phase of a civil DSS program.



# Chapter 5

## TECHSAT 21 MISSION CASE STUDY AND RESULTS

*One goal of future research should be to capture the pertinent physics of TechSat 21 in a quantitative analysis framework to enable a systematic understanding of how the key design variables impact mission performance as defined by end-user requirements. – The TechSat 21 Integrated Space Experiments Review Board*

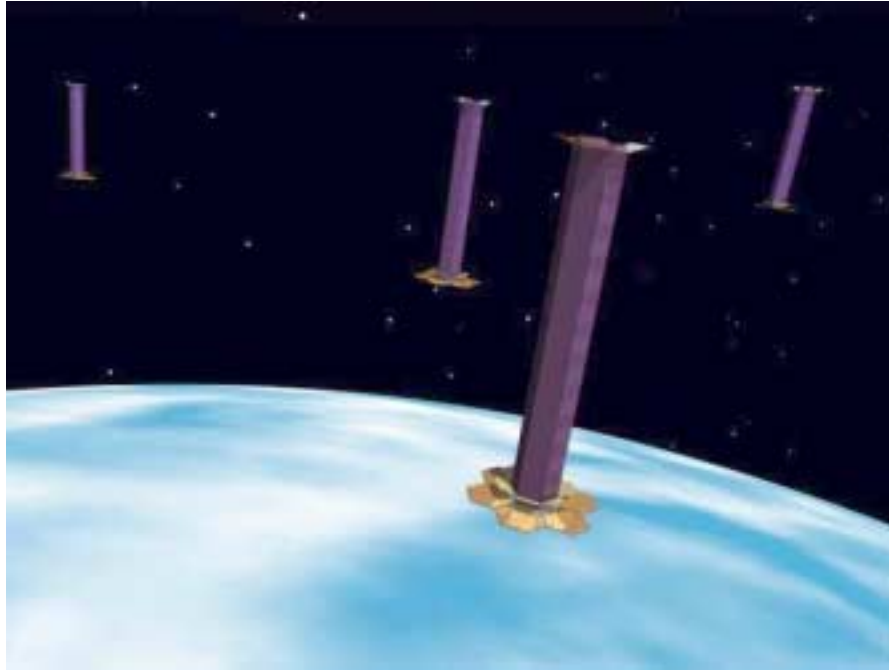
Chapter 5 presents a second class of space missions MMDOSA may be applied to – defense and national security missions. The military case study presented here differs from the civil case study presented in Chapter 4 in several important ways. First, the TechSat 21 trade space is much larger than the TPF trade space – so large in fact that it is not possible to completely enumerate every possible architecture for the TechSat 21 mission. Second, the multiobjective version of the TechSat 21 optimization problem contains more decision criteria (i.e. more dimensions to the objective function) than the equivalent TPF multiobjective optimization, and thus illustrates the ability of MMDOSA to find higher-order Pareto optimal sets of system architectures. Both of these factors combine to make the TechSat 21 conceptual design optimization problem more difficult than the TPF conceptual design optimization problem. This chapter demonstrates step-by-step the application of the MMDOSA methodology to the conceptual design of the TechSat 21 mission, and concludes by illustrating how MMDOSA found better system architectures than the existing point design architectures for this mission.

## 5.1 Introduction to the TechSat 21 Mission

Distributed satellite systems have been identified by the U.S. Air Force's New World Vistas Space Technology Panel as a revolutionary paradigm shift for future defense space missions [USAF, 1996]. A new concept for employing distribution within military space systems entails distributing the function of a single large satellite across a cluster of formation flying smaller satellites, analogous to dividing the functionality of a single mainframe computer across a network of PC workstations. The cluster in turn creates a "virtual satellite" connected via inter-satellite communication crosslinks and software. One might even imagine employing a constellation of these virtual satellites – that is a constellation of clusters to execute multiple missions. The potential advantages of distributed satellite systems and virtual satellites over traditional single satellite deployments for military missions include decentralization of resources, economies of scale, improved performance, and inherent adaptability. Appendix A provides a complete discussion on the benefits of DSS architectures. To test these ideas and validate the technologies required to make virtual satellites a reality, the Air Force Research Laboratory (AFRL) and the Air Force Office of Scientific Research (AFOSR) have initiated the Technology Satellite for the 21st Century (TechSat 21) program [Das & Cobb, 1998].

The TechSat 21 program has identified space-based radar as one mission that may significantly benefit from embracing the concept of virtual satellites as illustrated in Figure 5.1. The Air Force currently uses airborne radar platforms (JSTARS) to perform its Ground Moving Target Indication (GMTI) mission. While the potential advantages, including global coverage and large search areas, of moving radar to space-based platforms has long been recognized [Cantafio, 1989], space-based radar concepts have been hindered by the wide-diameter, large-mass, high-power, high-cost satellites required to provide the power-aperture product needed to overcome the large signal attenuation from free space loss. The AFRL proposes to use a distributed system of virtual satellites as an alternative to the traditional high cost space system architectures that have been previously proposed for the GMTI mission. The TechSat 21 program will demonstrate this concept [Das & Cobb,





**Figure 5.1** The TechSat 21 Formation-Flying Cluster Concept [Martin, 2000]

1998; Martin & Stallard, 1999], while simultaneously validating a number of technologies aimed at making military space systems smaller, cheaper, and more reliable [Wilson, 2000].

For the GMTI mission, the TechSat 21 system will employ formation-flying clusters of small satellites in low Earth orbit (LEO) using interferometric techniques (i.e. interferometric synthetic aperture radar (IFSAR)) to synthesize a much larger aperture than is contained on any single satellite. By using interferometric techniques, the cluster is able to synthesize the resolution of an antenna of dimension comparable to the size of the cluster itself (i.e. the baseline). The high angular resolution provided by this architecture offsets the disadvantage of large clutter spread arising from each individual satellite's small aperture, which in turn helps to resolve the ambiguity in Doppler between stationary ground clutter and moving targets. Table 5.1 compares the size and cost of the TechSat 21 concept with previous space-based radar GMTI concepts.

**TABLE 5.1** Space-Based Radar GMTI Concept Comparison [Martin, 2000]

<b>System Parameters</b>	<b>Rigid Panel Deployable</b>	<b>Integrated Antenna</b>	<b>Distributed Spacecraft (TechSat 21)</b>
Technology Freeze Date	1996	2003	2005
Satellite Mass	12,000 kg	4,400 kg	150 kg
Launch Vehicle	Titan IV	Atlas II	Taurus/Athena II
Lifecycle Cost (Normalized)	1.0	0.6	0.3

As one can see, the DSS concept appears to yield a two order-of-magnitude satellite mass savings and a factor of two to three lifecycle cost savings. However, several technological hurdles must be resolved to make a virtual satellite constellation a reality. These include cluster design [Sedwick et al, 1999], cluster maintenance against perturbative forces [Sedwick et al, 1998; Sedwick et al, 1999], signal processing [Hacker & Sedwick, 1999], distributed processing [Enright et al, 1999], collaborative behavior, and fault recovery. Each of these technologies is being developed, tested, and refined by the AFRL under the TechSat 21 program.

The AFRL believes the inherent adaptability present in a distributed architecture will enable the TechSat 21 system to perform other missions in addition to the GMTI mission. These additional missions include Air Moving Target Indication (AMTI), synthetic aperture radar imaging (SAR), passive radiometry (i.e. high accuracy geo-location), signal intelligence, and narrow beam/wide area coverage communication. Individual satellite clusters within the constellation can be rephased and reconfigured as needed by the Air Force to support these multiple missions. The case study presented in this chapter focuses solely on the GMTI mission for TechSat 21. The remainder of this chapter details how MMDOSA has been applied to the conceptual design of the TechSat 21 mission.

## **5.2 MMDOSA Step 1 – Create the GINA Model**

GINA – the Generalized Information Network Analysis methodology – is a systems engineering and architecting framework for the comparative analysis of distributed satellite

systems [Shaw et al, 2001]. GINA enables the comparison of different system architectures for a given mission. The foundation behind the GINA methodology is the assertion that most satellite systems are information disseminators that can be represented as information transfer networks [Shaw et al, 2000]. A summary of the steps in the GINA methodology is presented in Section 3.2. Through these steps, GINA allows the systems engineer to make meaningful, quantitative trades at the conceptual design level by directly relating lifecycle performance to lifecycle cost. The remainder of this section details the application of the GINA methodology to the AFRL's TechSat 21 GMTI mission.

### 5.2.1 GINA Step 1 – Definition of the Mission and CDP Objectives

In defining the mission objective, the systems engineer must explicitly state what the real application of the system is in terms of the user needs. To do this, the customer of the system must first be identified. The AFRL's TechSat 21 system has many stakeholders and potential customers within the U.S. Defense network. These customers include the Air Force, the intelligence community, and the remaining U.S. armed forces. For the case of TechSat 21, each customer values different products from the mission (ex. GMTI vs. geolocation), leading to the design of a multi-mission system.

Once the customer(s) has been identified, the mission objective may be defined. At this point it is important to distinguish between the TechSat 21 *experiment* and the TechSat 21 *operational system*. The objective of the TechSat 21 experiment, which will include two to four satellites deployed as a LEO cluster in 2004, is:

To explore, integrate, and demonstrate the basic technologies required to enable distributed satellite systems within the context of a space-based radar mission. [Martin & Stallard, 1999]

The objective of the TechSat 21 operational system, which in contrast to the TechSat 21 experiment, will entail an entire constellation of clusters is:

To provide the U.S. military with a global space presence for space-based radar (i.e. GMTI and possibly AMTI and SAR as well), geolocation, signal intelligence, and communication. [Martin, 2000]

Close inspection of this statement reveals what the system must accomplish to meet the needs of the user. First, the system must be capable of detecting, identifying, and tracking targets within the presence of signal clutter for the GMTI mission. Second, the system must provide enough angular resolution to geolocate objects within several meters, as might be required to find a downed pilot. Third, the system must be capable of passively receiving radio signals for signal intelligence. Fourth, the system must provide a fully connected network to enable global communication for the armed forces. Thus, defining the mission objective in terms of the user needs begins to elucidate the function of the space system to be designed. It also provides the top-level requirements of the system.

Next, the systems engineer must explicitly define the objective to be achieved during the conceptual design phase of the program to remove all ambiguity concerning what will have been accomplished by the end of this phase of the design process. For the TechSat 21 study, the CDP objective is:

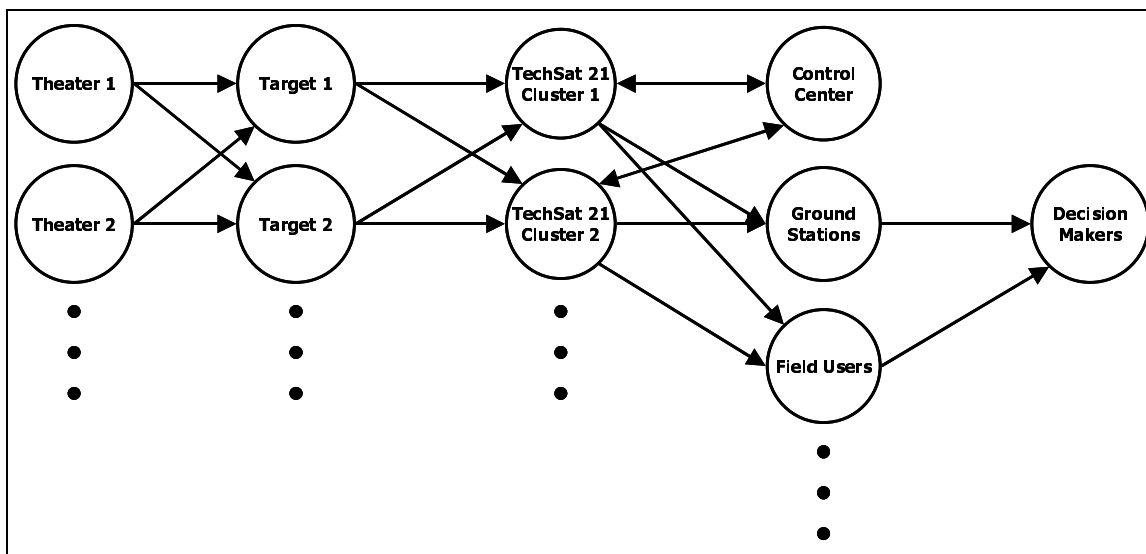
To enable and exercise the comparison of TechSat 21 conceptual design system architectures spanning the trade space, and implement as a software tool.

It is important to stress that the conceptual design phase goal is *not* to create a single point design, but rather to create a tool that may be used to model, analyze, and compare different system architectures for the AFRL's TechSat 21 GMTI mission.

### **5.2.2 GINA Step 2 – Information Network Representation**

Modeling the physical TechSat 21 system as an information network (Figure 5.2) entails defining all of the origin-destination (O-D) pairs in the system. For TechSat 21, the origin (i.e. source node(s)) are the theaters and their inherent targets of interest over which each satellite cluster passes, and the destination (i.e. sink node(s)) may be either decision makers in the Pentagon/Space Command or battle theater field users.

In the GINA methodology, the capability of an architecture is characterized by four quality of service parameters that relate to the detection process, and to the quantity, quality, and availability of the information that is processed through the network. These four



**Figure 5.2** The TechSat 21 System Modeled as a Network

parameters are signal isolation, information rate, information integrity, and the availability of these services over time [Shaw et al, 2000]. Once formulated, these four metrics serve as the minimum instantaneous capability requirements the system must meet to satisfy the customer.

- Isolation

Isolation refers to the ability of a system to isolate and distinguish information signals from different sources within the field of view. For the TechSat 21 GMTI mission, the minimum identifiable radar cross-section (RCS), which is determined by angular resolution via the maximum cluster baseline, determines the smallest sized targets that may be isolated and tracked by the system. Additionally, the minimum detection velocity (MDV) quantifies the slowest moving targets that may be isolated.

- Integrity

Integrity is a measure of the quality of the information being transferred through the network. In the case of the TechSat 21 GMTI mission, the integrity with which a target is identified is measured by the probability of detection, which is a function of the signal-to-noise ratio (SNR) and false alarm rate (FAR).

- Rate

Rate measures the speed at which the system transfers information between the sources and sinks in the network. In the TechSat 21 GMTI mission, the maximum revisit rate defines the longest time during which a particular the-

ater will not be in view of any satellite cluster. Within a theater currently in view, the target update time defines the rate at which the vector information for each target is renewed.

- Availability

Availability characterizes the instantaneous probability that information is being transferred through the network between all of the sources and sinks. For the TechSat 21 GMTI mission, this is the probability a target with a given RCS and MDV may be isolated and tracked under the required revisit and update rates with the desired probability of detection. The geometry of the constellation and design of the satellites affect the availability of the system.

At this point, the systems engineer must decide which of these capability parameters will become hard requirements that the system must meet to be considered feasible, and which parameters will serve to distinguish between system architectures. For TechSat 21, the isolation, rate, and availability parameters are hard requirements every potential design must meet, while the integrity will differ between architectures. Table 5.2 summarizes the capability parameters as applied to TechSat 21 along with the requirements placed on these parameters by the AFRL.

**TABLE 5.2** The GINA Capability Quality of Service Metrics for the TechSat 21 GMTI Mission

Capability	Equivalent Design Parameter	AFRL Requirement
Isolation	Minimum Radar Cross-Section	10 dbsm
	Minimum Detection Velocity	1.7 m/s
Integrity	Probability of Detection	Free Variable (Maximize)
Rate	Maximum Theater Revisit Time	15 min.
	Maximum Target Update Time	1 min.
Availability	% Time Meet Requirements	95%

### 5.2.3 GINA Step 3 – Develop System Metrics

To compare all the different architectures that are formulated during the conceptual design phase, GINA uses a quantifiable set of metrics – capability, performance, lifecycle cost, cost per function, and adaptability. The capability metrics have already been introduced in

Section 5.2.2. The remaining four metrics are discussed below. These are the metrics that will be used to evaluate and compare alternative TechSat 21 system architectures.

### **Performance**

While the four capability quality of service parameters measure how well a system architecture meets the capability requirements at any *instantaneous* point in time, the performance metric measures how well the architecture satisfies the mission requirements over the *entire* life of the mission, taking into account the potential degradation of satellites, clusters, and the constellation. For the TechSat 21 GMTI mission, the mean probability of detection at the desired availability during the nominal mission lifetime is the chosen performance metric.

To calculate this quantity, however, the additional complication of taking into account all of the possible failures that may occur within the system must be addressed. As individual satellites fail over time, the size and geometry of the TechSat 21 clusters will change to maintain the desired availability. In maintaining the desired availability, the system probability of detection will decrease as individual satellites within a cluster fail and the target signal-to-noise ratio decreases. Conversely, replenishing the constellation of clusters with new satellites over the course of the mission to replace failed satellites will boost the system probability of detection at the desired availability. Thus, the probability of detection integrity capability metric is a dynamic quantity that changes over time as a function of the number of operational satellites in each cluster. The number of operational satellites per cluster decreases when individual satellites fail and increases when replenishment satellites are launched.

To take into account potential failures and the effect they have on system performance, a reliability model is needed. GINA uses Markov reliability modeling techniques to determine both the probability that the system will continue to function over a given amount of time and the likelihood with which the system will function in different partially failed states throughout the mission. A Markov model must be created for each proposed Tech-

Sat 21 architecture. From the Markov model, a system of differential equations is developed, as shown in Eqn. 5.1, where  $\dot{\vec{P}}$  is the time rate of change of the state probability vector,  $A$  is the state coefficient matrix comprised of component failure rates, and  $\vec{P}$  is the state probability vector.

$$\dot{\vec{P}} = A\vec{P} \quad (5.1)$$

The solution to this set of linear, first order, partially coupled ordinary differential equations determines the probability of the TechSat 21 system being in any given state at a particular time. Coupling the outputs of the reliability model with the outputs of the capability model (i.e. the target probability of detection in each operational state) yields the total performance of the system. This coupling equation is the utility function. Dividing the utility function by the mission duration yields the mean target probability of detection over the course of the mission.

$$P_d(T) = \frac{\int_0^T \sum_{i=1}^n C_i P_i(t) dt}{T} \quad (5.2)$$

where for TechSat 21  $P_d$  is the mean probability of detection at the desired availability during the nominal mission lifetime,  $T$  is the nominal mission lifetime,  $n$  is the total number of operational states (ex. state 1 = 8 out of 8 satellites operational per cluster, state 2 = 7 out of 8 satellites operational per cluster, etc.),  $C_i$  is the capability (i.e. instantaneous probability of detection) in each state  $i$ , and  $P_i(t)$  is the probability of being in each operational state  $i$  as a function of time  $t$ .

### Lifecycle Cost

Four primary components of the lifecycle cost model are the payload cost, satellite bus cost, launch cost, and operations cost. The payload and satellite bus theoretical first unit (TFU) costs are computed via The Aerospace Corporation Small Satellite Cost Model



(SSCM) cost estimation relationships (CERs) in Table 3.4 and Table 3.5, and then a learning curve is applied to compute the total cost in constant year dollars as a function of the required number of units for a particular system architecture. The launch cost model is divided into initial deployment costs and replenishment costs. The initial deployment cost model uses the optimization algorithm in Appendix B to select a suite of launch vehicles from the complete set being considered by the AFRL that can deploy all of the satellites into their required orbital slots at minimum cost. Inputs to the deployment launch cost model are the total number of satellites in the TechSat 21 constellation, the mass and volume of the satellites, the orbital elements of the constellation, and the performance parameters of each potential launch vehicle. The replenishment launch cost model uses the same optimization algorithm as the initial deployment cost model, with the total number of satellites in the TechSat 21 constellation input being replaced with the expected number of satellite failures over time (i.e. the number of satellites that will need to be replenished) as computed by the TechSat 21 Markov model. Finally, the operations cost model estimates the operations crew size, operations crew cost, flight/ground software size, software costs, and total operations lifecycle cost of operating the TechSat 21 constellation over the mission duration.

### **Cost Per Function**

The cost per function (CPF) metric provides a measure of the cost of an architecture versus its performance [Shaw et al, 2001]. For TechSat 21, the CPF metric may be defined two ways – cost per protected square kilometer of theater and cost per unit probability of detection at a fixed availability. Cost per protected square kilometer of theater is calculated by dividing the estimated lifecycle cost of the architecture by the total Earth surface area the constellation may view. Assuming each constellation provides global coverage, this first definition of CPF reduces to lifecycle cost. Cost per unit probability of detection at a fixed availability is calculated by dividing the estimated lifecycle cost of the architecture by the system mean probability of detection as computed by the performance model.

### **Adaptability**

Adaptability measures how flexible an architecture is to changes in design assumptions and mission requirements [Saleh, 2002]. In one sense, adaptability may be thought of as the sensitivity or elasticity of the CPF of a particular architecture to incremental changes in an assumption or requirement. For the TechSat 21 mission, potential assumptions that could be altered to measure architecture sensitivity include radar processing algorithms, learning curve slopes, and component reliabilities. In another sense, adaptability may be thought of as the flexibility of a particular architecture to a new set of mission requirements. This version of adaptability becomes especially important when considering the potential multi-mission nature of the TechSat 21 constellation. An example of flexibility for TechSat 21 is the ability of an architecture to transition from a GMTI mission to a geolocation mission, or from a SAR mission to a signal intelligence mission.

#### **5.2.4 GINA Step 4 – Partition the Conceptual Design Problem**

The next step in the GINA process involves listing all of the design variables that the systems engineer has control over. From this list, key independent variables that drive the design of the system are extracted and inserted into the design vector. The design vector contains the attributes that will distinguish and differentiate alternative system architectures. Remaining variables are placed in the constants vector. The constants vector contains attributes that the systems engineer has control over, but that will not distinguish or differentiate alternative system architectures. For example, the assumed areal density of the satellite solar arrays is set to a constant value as a function of the type of solar array (ex.  $1 \text{ kg/m}^2$  for gallium arsenide multi-junction cells) regardless of the other attributes of the architecture.

For TechSat 21, six variables – the number of spacecraft per cluster, radar antenna aperture diameter, radar payload power, constellation altitude, number of clusters per orbital plane, and number of orbital planes in the constellation – make up the TechSat 21 design vector. Table 5.3 lists the range of values considered for each design vector variable in the

TechSat 21 case study. Each design vector defines a unique TechSat 21 system architecture.

**TABLE 5.3** The TechSat 21 Design Vector

Design Vector Variable	$\Gamma$	Nomenclature	Range	Discretization
# Satellites Per Cluster	$\gamma_1$	$n_{scl}$	4 - 16	1
Aperture Diameter	$\gamma_2$	$D_{ant}$	0.5 m - 4.0 m	0.5 m
Radar Payload Power	$\gamma_3$	$P_t$	100 W - 1000 W	100 W
Constellation Altitude	$\gamma_4$	$h$	500 km - 1500 km	100 km
# Clusters Per Orbital Plane	$\gamma_5$	$n_{clp}$	3 - 10	1
# Orbital Planes	$\gamma_6$	$n_{op}$	3 - 10	1
<b>Total # Permutations</b>			<b>732,160</b>	

Once the design vector has been identified, it may be matrixed against the GINA system metrics. The chart in Figure 5.3 establishes the relationship between the design vector (i.e. top row) and the metrics (i.e. left column) by which competing architectures will be judged. Notice the fundamental relationships between the elements of the design vector and the GINA metrics. For example, the altitude of the constellation directly determines the theater revisit rate. Hence, the constellation altitude drives the revisit rate metric.

Once this matrix has been completed, different attributes can be aggregated into groups of modeling needs that allow the recognition of important differences between competing architectures. The systems engineer may now define the modules that will be required to develop, analyze, and compare different architectures for the TechSat 21 mission. These modules capture the fundamental physics, costs, and systems engineering trades of the TechSat 21 conceptual design problem. For TechSat 21, six core modules are required:

1. Constellation module
2. Radar module
3. Payload module
4. Satellite Bus module
5. Launch & Operations module

Top Trades Capability Metrics	Number of Satellites	Number of Clusters	Altitude	Aperture Diameter	Transmission Power	Autonomy
Isolation	N/A	N/A	MDV and maximum cluster baseline	N/A	N/A	N/A
Rate	N/A	Minimum revisit time	Revisit time and area search rate	Footprint coverage area	Required dwell time	N/A
Integrity	Antenna gain pattern from cluster projection	N/A	Received signal strength	Mainlobe vs. sidelobe gain	Impacts the signal-to-noise ratio	N/A
Availability	Overall system reliability	Overall system reliability	N/A	N/A	Time required to scan a target area	Recovery from non-permanent failure
Cost per Function	Learning curve effects	Learning curve effects	Launch, eclipse, & drag	Antenna cost	Size of spacecraft	Software development and ops costs

Constellation	Systems Analysis	Launch/Operations	Radar	Bus	Payload

Figure 5.3 TechSat 21 Model Component Identification Metrics Matrix

### 6. Systems Analysis module

Detailed analysis of the model component identification metrics matrix begins to yield insight to the trade space. Figure 5.4 illustrates an alternative view of the TechSat 21 trade space. Each blue arrow represents an element of the design vector, while the green and red boxes represent the benefit in terms of the GINA metrics of increasing or decreasing the value of that parameter. For example, increasing the number of spacecraft per cluster improves the integrity and availability capabilities of the system, but at the expense of launch costs and operations costs. Likewise, decreasing the total number of clusters in the constellation lowers the lifecycle cost, but at the expense of the theater revisit rate. Similar trade-offs exist for each element of the design vector. What makes the TechSat 21 conceptual design problem so difficult is that all of these trades are coupled in a nonlinear fashion. One cannot simply optimize each element of the design vector in isolation (i.e. choose the number of clusters that maximizes revisit rate, choose the number of satellites per cluster that maximizes SNR, etc.) from each other because each element of the design

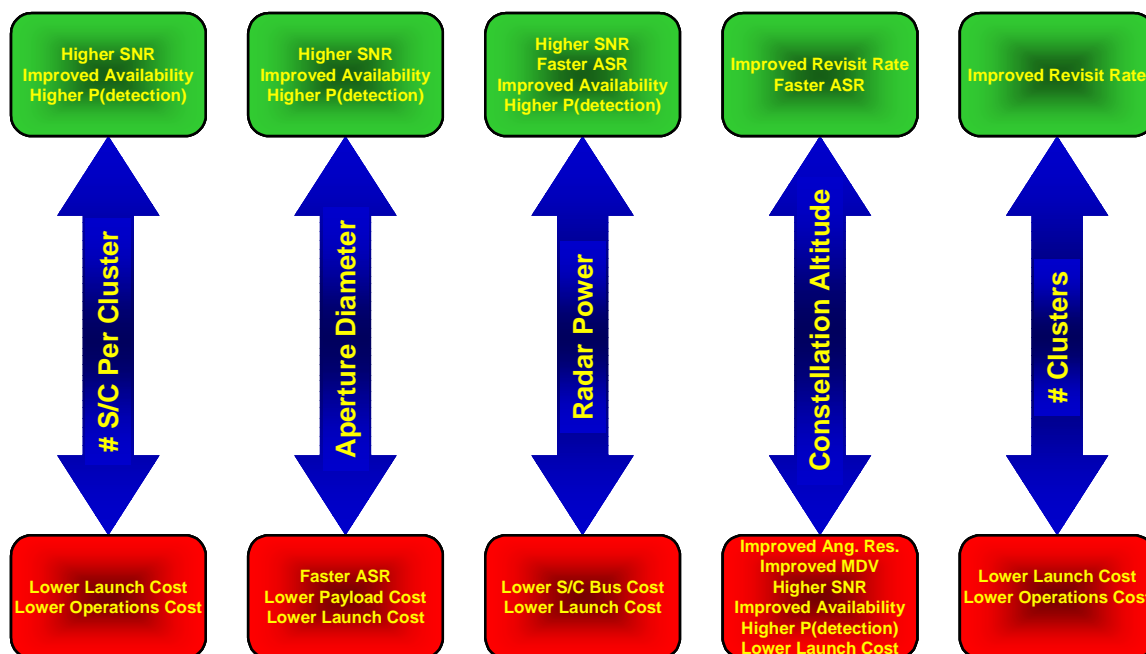


Figure 5.4 Alternative View of the TechSat 21 Trade Space

vector impacts all elements of the design. This is where the true strength of the MMDOSA methodology demonstrates itself, as MMDOSA enables the systems engineer to capture quantitatively all these couplings and then search the nonconvex trade space for the most cost-effective families of system architectures.

### 5.2.5 GINA Step 5 – Develop Simulation Software

A module is a Matlab m-file that models a specific subsystem or function within the mission. The purpose of the modules is to develop a tool that sizes the subsystems within an architecture for a given set of inputs. If implemented correctly, the modules should also aid the systems analysis methodology by computing the metrics for a given architecture. This allows for rapid evaluations and comparisons of alternative DSS architectures.

Figure 5.5 illustrates the Matlab modules identified in the model component identification metrics matrix for the TechSat 21 GINA model. The two sets of inputs to the model are the design vector and constants vector. The constellation module propagates the orbits of each satellite cluster to compute the coverage statistics of the constellation. Using these

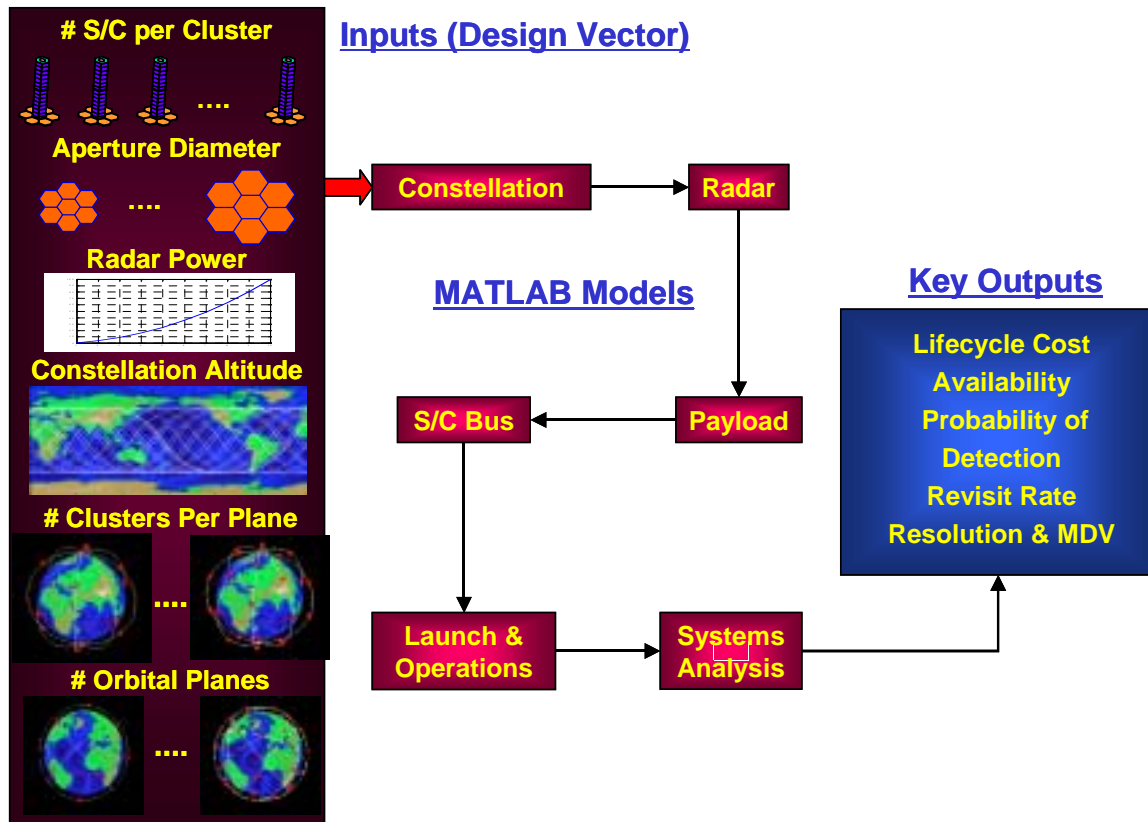


Figure 5.5 TechSat 21 GINA Model Simulation Software Block Diagram

statistics in combination with the design vector inputs, the radar module calculates the expected performance of the radar system (i.e. probability of detection, availability, etc.). The payload module then sizes the radar transmit/receive antenna and computes its theoretical first unit cost. Once the payload has been sized, the spacecraft bus module sizes and costs the satellite bus required to support the payload. The launch and operations module then calculates the initial deployment cost, replenishment cost, and annual operations cost for supporting the constellation. Finally, the systems analysis module couples the performance, reliability, and lifecycle cost analyses. Outputs of the GINA TechSat 21 simulation model include detailed subsystem specifications along with the GINA capability, performance, lifecycle cost, and cost per function metrics.

The mathematical models within each module can become intricate, making the consistent management of information within the model a complex issue. An  $N^2$  diagram is used as

the interface control document during this step. Figure 5.6 illustrates the TechSat 21 GINA model  $N^2$  diagram. As one can see, certain modules are further decomposed into submodules. This modular division of the TechSat 21 conceptual design process reduces software development risk by reducing coupling and simplifies the simulation code development as each module is separately testable.

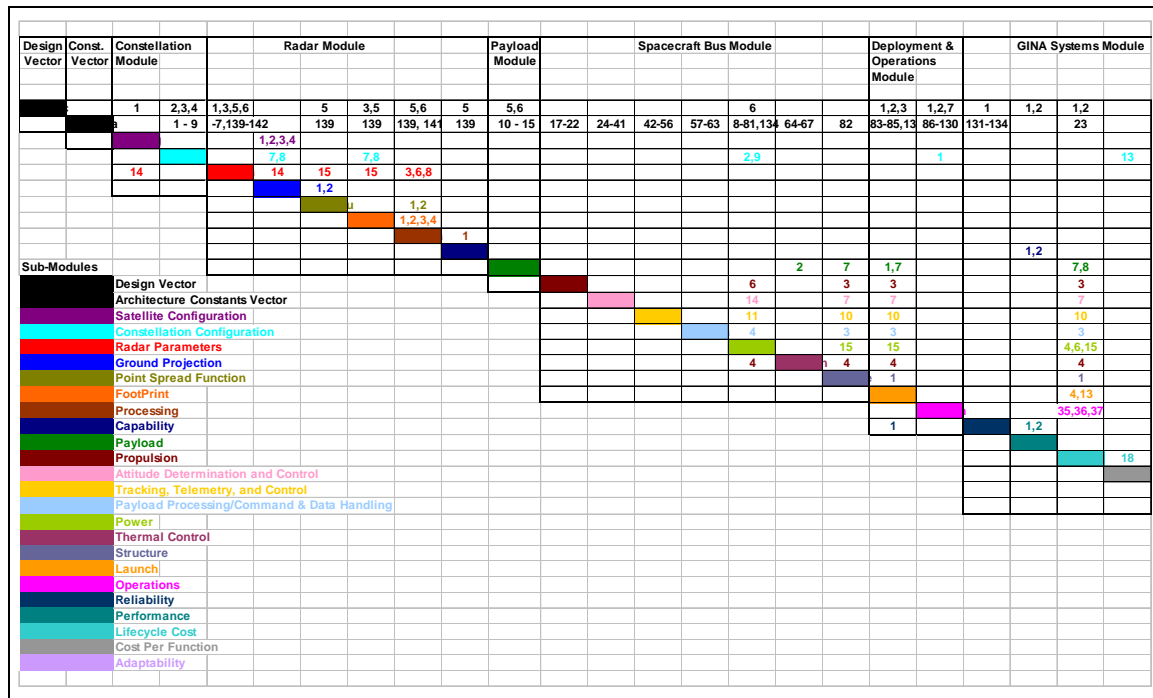


Figure 5.6  $N^2$  Diagram for the TechSat 21 GINA Model

An  $N^2$  diagram, also known as a Design Structure Matrix [Browning, 1998], is an  $N \times N$  matrix used by systems engineers to develop and organize interface information [Kockler, F.R., 1990]. The submodules (i.e. Matlab m-file functions) are located along the diagonal of the matrix. Each color-coded number designates a unique variable within the submodule. Inputs to each submodule are vertical and outputs are horizontal. Aggregation of the submodules into complete modules is illustrated by the black boxes enveloping different sections of the diagonal. The  $N^2$  diagram in Figure 5.6 provides a visual representation of the flow of information through the conceptual design process and is used to connect all of

the Matlab functions to enable an automated simulation of different TechSat 21 architectures.

### Constellation Module

Figure 5.7 illustrates key inputs and outputs of the constellation module. With knowledge of the constellation altitude and geometry from the design vector in combination with satellite viewing angle constraints from the constants vector, a set of constellation coverage statistics are computed for use by other modules downstream in the TechSat 21 GINA model. Key outputs from this module include average, minimum, and maximum theater revisit times; elevation and azimuth angle statistics; and local atmospheric environment properties.

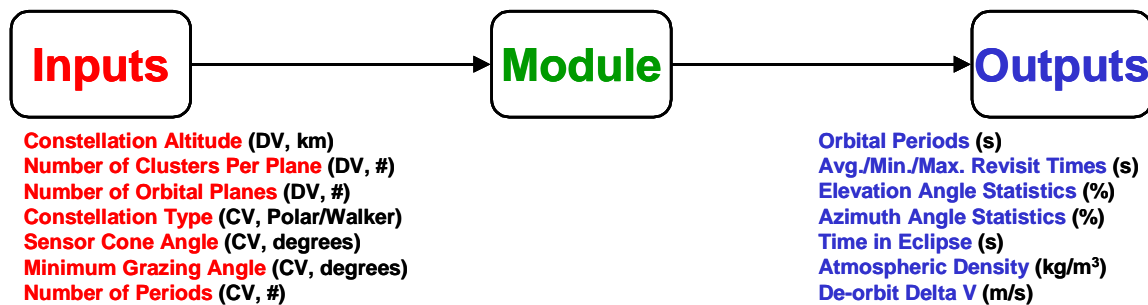


Figure 5.7 Constellation Module Inputs and Outputs

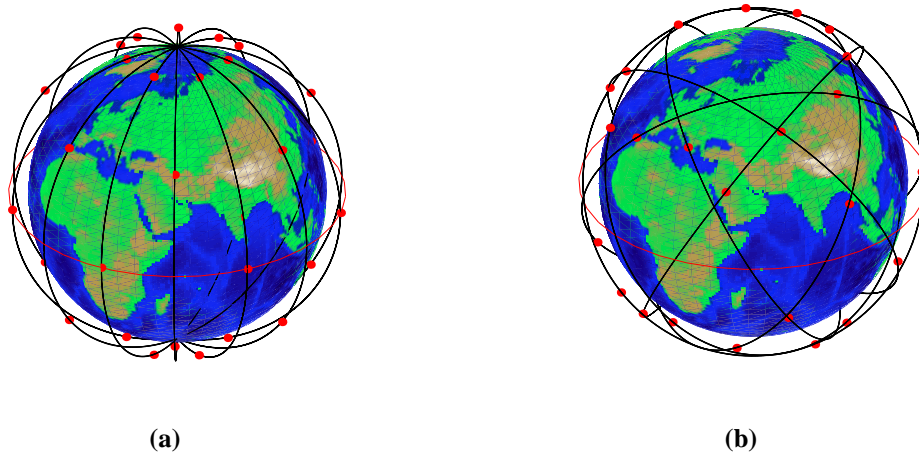
The full set of orbital elements may be derived for each satellite cluster in the TechSat 21 constellation from the design and constants vectors alone. Table 5.4 lists the orbital elements and their source for the two types of constellations that may be automatically generated, propagated, and analyzed by the TechSat 21 GINA simulation model. These two types of constellations are polar constellations (Figure 5.8a) and Walker constellations (Figure 5.8b).

The period  $P$  of each satellite may be calculated as a function of the constellation altitude  $h$  designated in the design vector:



**TABLE 5.4** Polar and Walker Constellation Orbital Elements for the TechSat 21 GINA Model

Orbital Elements	Polar Constellation	Walker Constellation
Semi-Major Axis ( $a$ )	DV	DV
Eccentricity ( $\epsilon$ )	0	0
Inclination ( $i$ )	90°	CV
Longitude of the Ascending Node ( $\Omega$ )	DV	DV
Argument of Periapsis ( $\omega$ )	undefined	undefined
True Anomaly ( $v$ )	DV, CV	DV, CV

**Figure 5.8** Sample Polar (a) and Walker (b) TechSat 21 Constellations Generated by the GINA Model

$$P = 2\pi \sqrt{\frac{(R_e + h)^3}{\mu_e}} \quad (5.3)$$

where  $R_e$  is the radius of the Earth and  $\mu_e$  is the gravitational constant of the Earth.

With knowledge of the orbital elements and period of each satellite cluster, the average, maximum, and minimum theater revisit times may be computed. First, a latitude-longitude grid is placed over the surface of the Earth. The resolution of this grid is defined in the constants vector. Intersections between each latitude and longitude line in this grid represent theaters of interest for the TechSat 21 constellation. The GINA model propagates the orbit of each cluster in the TechSat 21 constellation over the Earth for the number of orbital periods specified in the constants vector at the time-step resolution

designated in the constants vector. In this manner, ground tracks for all of the clusters in the constellation are created (Figure 5.9a). At each time-step in the orbit propagation simulation, the number of satellite clusters in view of each theater of interest is calculated as a function of the sensor cone and minimum grazing angles defined in the constants vector. Aggregating this information for all the theaters of interest over the entire simulation yields the average, minimum, and maximum theater revisit times for a TechSat 21 constellation. Figure 5.9b illustrates a sample plot that may be created from this data – maximum revisit time as a function of latitude bands for a particular Walker constellation.

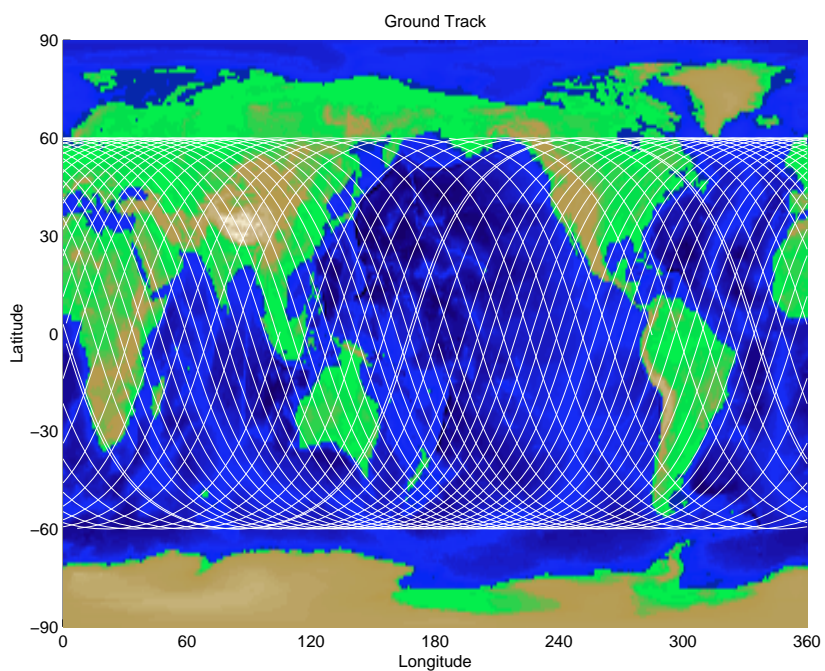
By keeping track of the elevation angle and azimuth angle between each theater of interest and each TechSat 21 cluster in view across every time-step in the simulation, the probability density functions of the maximum elevation angle and azimuth angle can be computed between the constellation and any theater of interest. The statistics from these probability density functions are used downstream in the GINA model to assess the performance of the GMTI radar mission. Figure 5.10 illustrates sample probability density functions for maximum elevation angle (Figure 5.10a) and azimuth angle (Figure 5.10b) as a function of longitude at the equator.

The maximum amount of time  $t_e$  each TechSat 21 satellite remains eclipsed from the sun by the Earth is computed as:

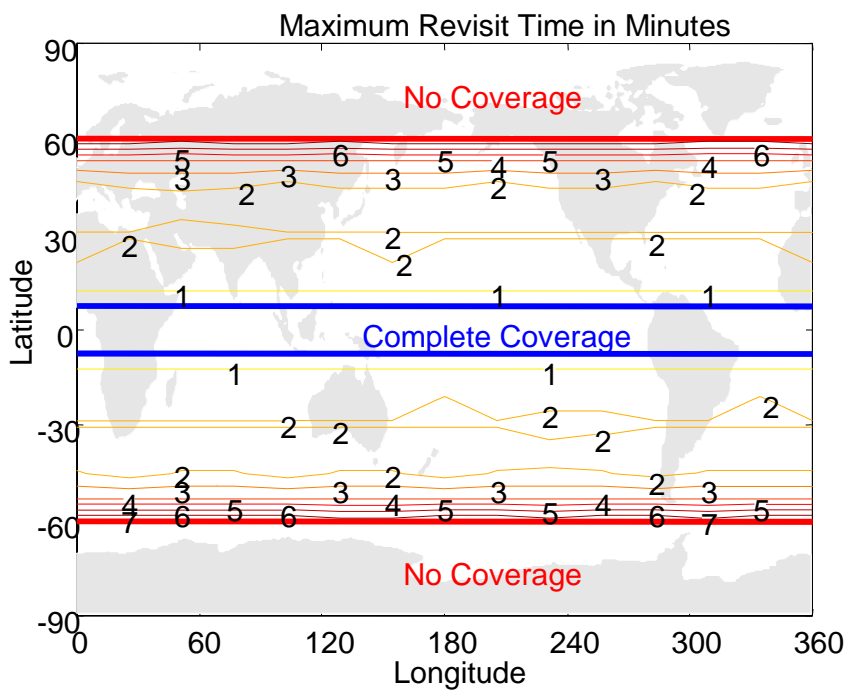
$$t_e = P \left( \frac{\text{asin}\left(\frac{R_e}{a}\right)}{\pi} \right) \quad (5.4)$$

This data gets used downstream in the GINA model by the spacecraft bus module to size the batteries in the power subsystem of each satellite.

The atmospheric density within the local environment of each TechSat 21 satellite is estimated from a look-up table of the Mass Spectrometer Incoherent Scatter atmospheric model as a function of solar activity [Larson & Wertz, 1992]. This information is later

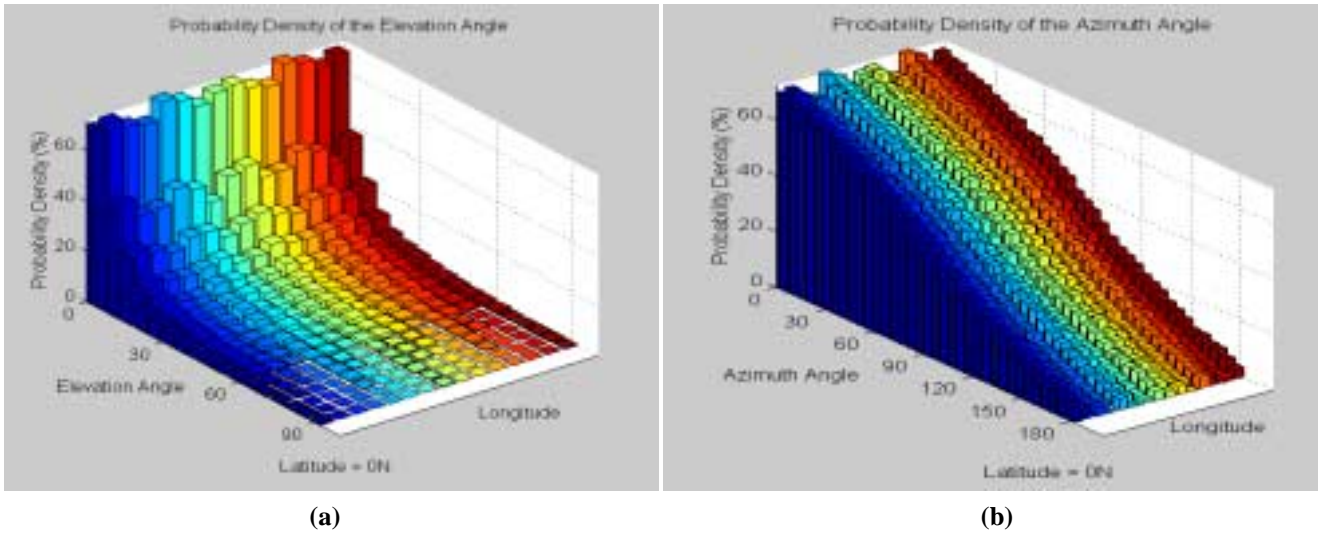


(a)



(b)

**Figure 5.9** Sample TechSat 21 GINA Model Ground Track (a) and Maximum Revisit Time (b) Plots



**Figure 5.10** Maximum Elevation Angle (a) and Azimuth Angle (b) Probability Density Functions at the Equator for a Sample TechSat 21 Constellation

used within the spacecraft bus module in the GINA model to compute the  $\Delta V$  corresponding fuel requirements for orbit maintenance over the life of each TechSat 21 satellite.

The GINA methodology models the entire lifecycle of the space system, including system disposal. Disposal options for satellites include natural atmospheric decay and reentry, boosts to a benign orbit, and propulsion induced atmospheric reentry. Because the TechSat 21 satellites will operate in LEO, they will be disposed of via propulsively induced controlled atmospheric reentry. The  $\Delta V_{do}$  required to induce reentry may be calculated as:

$$\Delta V_{do} = V \left( 1 - \sqrt{\frac{2(R_e + h_{do})}{R_e + a}} \right) \quad (5.5)$$

where  $V$  is the velocity of the satellite and  $h_{do}$  is the altitude to which the satellite perigee is lowered by the propulsive burn to induce atmospheric reentry.  $\Delta V_{do}$  is used downstream in the GINA model by the propulsion submodule within the spacecraft bus module to compute the reserve fuel load required aboard each TechSat 21 satellite for safe deorbit.

## Radar Module

Figure 5.11 illustrates the key inputs and outputs of the radar module. It should be repeated that the GINA TechSat 21 radar module only models the GMTI radar mission, and not the SAR or MTI missions. With knowledge of the number of satellites per cluster, constellation altitude, and the radar antenna parameters from the design vector in combination with a host of radar signal processing parameters from the constants vector and the coverage statistics from the constellation module; a set of radar target detection and tracking statistics are computed to assess the GMTI performance of the TechSat 21 system architecture. Key outputs include cluster footprint, minimum detectable velocity, and probability of detection statistics. This section provides a cursory overview of the radar module as excerpted from the reference by Hacker et al. For greater detail on the radar module, including all of the formulas and computer code, please refer to the report on this subject by Hacker et al [Hacker et al, 2000].

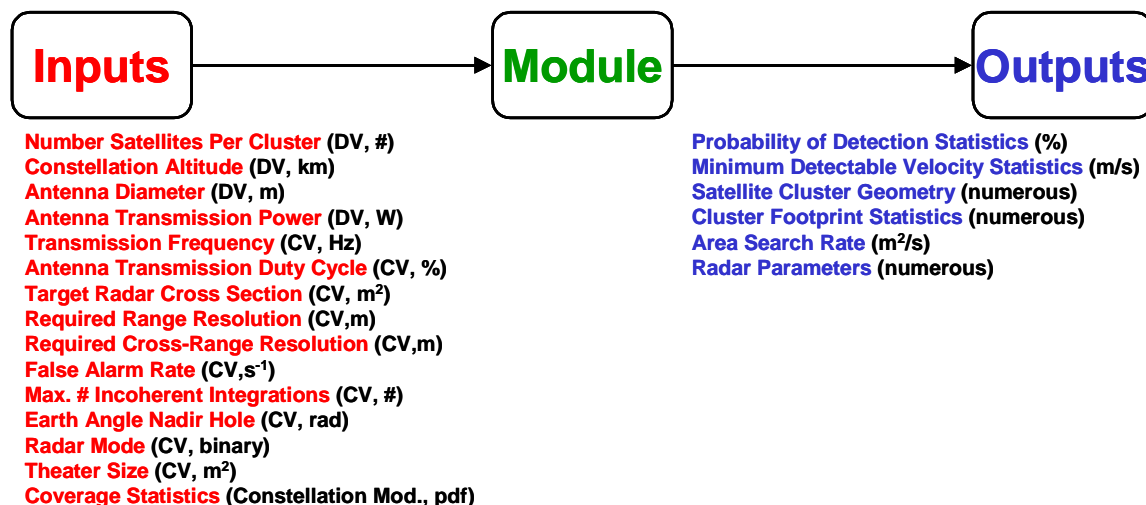


Figure 5.11 Radar Module Inputs and Outputs

Most radar systems use signals transmitted and received by a single antenna to detect targets. In such systems, a large antenna is required to determine a target's location with a high degree of accuracy. This is especially true for space-based systems, where the antenna can be located hundreds of kilometers from the target. The requirement for a

large antenna drives up the size of the spacecraft bus and thus the size and cost of the launch vehicle(s) required to deploy the system. For these reasons, single aperture space-based radar appears to be prohibitively expensive.

An alternative approach locates multiple apertures on individual satellites flying in formation to create a large baseline sparse array in which the signals received from the ground are combined via interferometry [Hacker & Sedwick, 1999]. Such a configuration utilizes small antennas with the long baseline combined signals to provide high resolution. This system has the potential to lower lifecycle cost by taking advantage of using smaller satellites at lower orbital altitudes. The primary source of the cost savings is the decoupling of the radar system angular resolution from the satellite aperture diameter provided by interferometry. Costs are further lowered by allowing the satellites in a single cluster to share the computational burden of processing the radar signals, reducing the required processor capability.

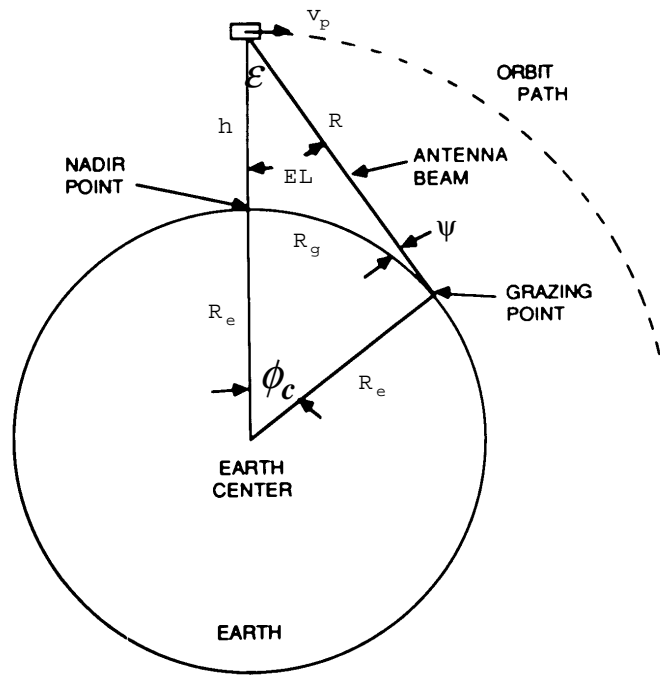
A major challenge in space-based radar interferometry involves separating the received target signals from the received clutter signals [Murray et al, 1997]. Possible signal processing approaches for achieving the required clutter suppression include displaced phased center antenna (DPCA) processing [Cantafio, 1989; Skolnik, 1980], space-time adaptive processing (STAP) [Klemm, 1998; Rabideau & Kogon, 1999], and deconvolution [Marais & Sedwick, 2001]. While DPCA processing is the easiest to implement, it suffers from the tight constraints placed on the motion of the spacecraft, the limits placed on the pulse repetition frequency (PRF) by the antenna platform velocity, and the fact that only a fraction of the aperture is used to receive the signal at any one time. STAP overcomes these problems by taking a series of temporal taps based on the PRF for each subarray of the radar antenna to formulate the covariance matrix of the interfering sources and then calculating the matrix inverse. Transformation of this matrix to the space-time domain gives the angular locations and Doppler spectra of the received clutter. However, real-time implementation of this signal processing method is computationally expensive. Additional drawbacks to STAP include the need for detailed clutter models, the limit on

the number of interference sources that can be eliminated as a function of the number of apertures in a cluster, and the constraints placed on the cluster geometry to maintain an adequate distribution of apertures. Deconvolution signal processing, a newer technique, takes advantage of the spatial diversity of the apertures to cancel the entire clutter return in the received radar signals. The improved target probability of detection of this signal processing algorithm over conventional STAP results from the fact that noise becomes the only source of interference in the system when the target and clutter returns are separated. Because of the potential performance increase, the TechSat 21 radar module employs a deconvolution signal algorithm for the GMTI mission. For more information on the deconvolution algorithm please see the reference by Marais and Sedwick [Marais & Sedwick, 2001].

The primary components of the TechSat 21 radar module are the radar footprint model and the coverage area model. The footprint model calculates the received antenna pattern based on the projection of the individual satellite locations on the ground and combines this antenna pattern with the properties of the footprint to evaluate the probability of target detection in the presence of noise and clutter. Key components of this model include the calculation of:

1. Radar Geometry
2. Maximum Detection Range
3. Satellite Cluster Baseline
4. Pulse Properties
5. Satellite Configuration
6. Transmit and Receive Gain
7. Clutter Doppler Profile
8. Clutter Cross Section
9. Received Signal Processing
10. Isolation Capability
11. Overall Radar System Performance.

Figure 5.12 illustrates the geometrical parameters between satellite clusters and the Earth in the TechSat 21 radar module.



**Figure 5.12** Radar-Earth Geometry [Cantafio, 1989]

The parameters radar range to target, grazing angle of the footprint, and footprint area characterize the geometry of the GMTI mission. The target range  $R$  is the distance from a TechSat 21 cluster to the location on the ground where the footprint is projected, and may be computed as:

$$R = \sqrt{(R_e + h)^2 - 2R_e(R_e + h)\cos\phi_c + R_e^2} = R_e \frac{\sin\phi_c}{\sin\epsilon} \quad (5.6)$$

where  $\phi_c$  is the Earth angle and  $\epsilon$  is the elevation angle. The elevation angle is computed as:

$$\epsilon = \text{asin}\left(\frac{R_e \sin\phi_c}{R}\right) \quad (5.7)$$



The grazing angle  $\psi$  of the footprint corrects the elevation angle of the illuminated ground location to account for the curvature of the Earth:

$$\psi = \arccos\left(\sin \varepsilon \frac{(R_e + h)}{R_e}\right) = \frac{\pi}{2} - \varepsilon - \phi_c \tag{5.8}$$

The footprint area  $A_f$  is proportional to the antenna beamwidth  $\Delta\theta$  elongated by the Earth’s curvature:

$$A_f \cong \frac{\Delta\theta^2 R^2}{\sin \psi} \tag{5.9}$$

The maximum distance at which a space-based radar system can detect targets depends upon either the distance to the horizon, the maximum off-boresight look angle for the electronic antenna, or the minimum SNR limit [Hacker et al, 2000]. The absolute maximum range  $R_{max}$  of the radar is a function of the orbital altitude [Cantafio, 1989]:

$$R_{max} = \sqrt{(R_e + h)^2 - R_e^2} \tag{5.10}$$

For space-based radar, the projected length of the antenna is shortened by the cosine of the look angle as the antenna beam is steered away from nadir. This restricts the maximum off-boresight angle to  $\pm 60^\circ$  [Stimson, 1998]. The minimum SNR limit is a function of the number of integrated pulses and the false alarm probability.

The satellite cluster baseline defines the maximum separation distance between individual apertures in a single TechSat 21 cluster, and represents the synthesized aperture that results from the combination of the received signals by the sparse apertures located on each satellite [Hacker et al, 2000]. The minimum cluster baseline may be determined in two ways. One method entails sizing the cluster baseline to achieve the required cross-range resolution at the maximum detection range. The second method involves using the required MDV to compute a lower bound on the cluster baseline to ensure detection of slow-moving targets. In the radar module, the satellite cluster minimum baseline is set as the larger result of these two computations.

Computed pulse properties include the pulse repetition frequency (PRF) and area search rate (ASR). The PRF is selected to ensure that no range ambiguities exist in the footprint:

$$PRF_{\max} = \frac{c}{2L_{rf}} \sec \psi \quad (5.11)$$

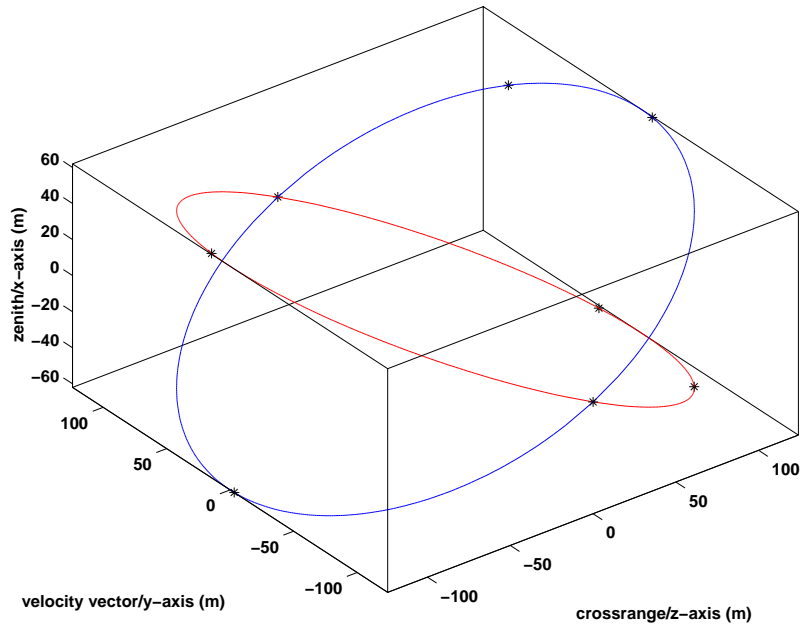
where  $c$  is the speed of light,  $L_{rf}$  is the length of the footprint, and  $\psi$  is the grazing angle. The ASR in search mode based on a continuous search of the field of regard by a single cluster is:

$$ASR_{\text{search}} = 2v_p R_{\max} \sin \epsilon_{\max} \quad (5.12)$$

where  $v_p$  denotes orbital (i.e. radar aperture platform) velocity and  $\epsilon_{\max}$  is the maximum elevation angle.

The configuration of satellites within a TechSat 21 cluster must maintain enough spatial diversity of the apertures to provide sufficient target detection capability in all directions [Hacker et al, 2000]. In the TechSat 21 radar module, this is done by modifying existing non-redundant baseline configurations [Leech, 1956] so that the distribution of satellites throughout their entire orbital period never degenerates to a single line, thus ensuring that the nominal spatial separation required for processing the received radar signals is always maintained [Sedwick et al, 1999]. An example of such a configuration for an eight-satellite cluster is illustrated in Figure 5.13.

Once the satellite configuration within a single cluster has been determined, the radar module proceeds to compute the transmit and receive gains for the sparse aperture based on the projection of the satellite positions at the ground location of interest [Hacker et al, 2000]. The effective area of the aperture during transmission and reception determines the overall gain pattern of the signal reflections that arrive at a radar antenna. In a single aperture radar system, these gains are identical. This is not the case, however, for a multiaperture system like TechSat 21. While the transmission gains will be the same, the receive gain will depend on the characteristics of the synthesized aperture. From the angular loca-



**Figure 5.13** Satellite Orbits in Hill’s Frame for an Eight-Satellite TechSat 21 Cluster [Hacker et al, 2000]

tion of the footprint on the ground, the radar module computes the receive gain  $G$  of a single TechSat 21 cluster by projecting the satellite positions in Hill’s reference frame to the location of a particular point in the footprint via the following two transformation matrices:

$$G_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{bmatrix} \tag{5.13}$$

$$G_z = \begin{bmatrix} \cos\left(\frac{\pi}{2} - \epsilon\right) & \sin\left(\frac{\pi}{2} - \epsilon\right) & 0 \\ -\sin\left(\frac{\pi}{2} - \epsilon\right) & \cos\left(\frac{\pi}{2} - \epsilon\right) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.14}$$

where  $\alpha$  is the azimuth angle. The radar module uses this single cluster ground projection to calculate the array factor gain, which is the energy increase in the processed signals from separated spacecraft interferometry. Computing the product of the single aperture gain and array factor gain yields the complete receive gain  $G_{\text{int}}$  of the interferometric array, also known as the point spread function:

$$G_{\text{int}} = \left[ \left( \frac{\pi(1 + \cos\theta)D_{\text{ant}}}{\lambda} \right) \left( \frac{J_1\left(\frac{\pi D_{\text{ant}} \sin\theta}{\lambda}\right)}{\frac{\pi D_{\text{ant}} \sin\theta}{\lambda}} \right) \right] \left| \sum_{n=1}^N \exp\left(\left(-\frac{2\pi i}{\lambda}\right)(\vartheta_i x_n + \vartheta_j y_n)\right) \right|^2 \quad (5.15)$$

where  $\theta$  is the off-boresight angle,  $D_{\text{ant}}$  is the diameter of a single aperture,  $\lambda$  is the wavelength of the radar signals,  $\vartheta_i$  &  $\vartheta_j$  are the image angular coordinates, and  $x_n$  &  $y_n$  are the coordinates of the physical projection of the  $n$ -th aperture position in the array.

Both the transmit and receive gains influence the strength of the signal reflections from the desired target and the undesired clutter. Clutter is defined as the reflected signals from everything in the footprint other than the target (i.e. ground, trees, water, etc.). Thus, in order to estimate properly the target detection capability of a TechSat 21 architecture, the radar module must first compute the properties of the ground clutter within a footprint. These clutter properties include the clutter Doppler profile and the clutter cross-section. The received signals in a space-based radar system are shifted and spread in frequency due to the platform motion of the TechSat 21 cluster. The clutter signals are blue-shifted in the direction of the cluster motion and red-shifted in the opposite direction according to the following equation:

$$f_D = \frac{2v_p}{\lambda} \sin\epsilon \cos\alpha \quad (5.16)$$

where  $f_D$  is the Doppler shift in Hertz,  $v_p$  is the orbital (i.e. radar aperture platform) velocity, and  $\alpha$  is the azimuth angle measured from the velocity vector. Since each cell in the footprint contains a finite area, the Doppler shift of the clutter in each cell is spread over a range of frequencies. This leads the clutter return from a single cell to fall in multiple

Doppler bins. The resulting Doppler spread is a function of the beamwidth of the synthesized aperture and may be computed by taking the differential of Eqn. 5.16:

$$\Delta f_D = \frac{4v_p}{B} \sqrt{(\sin \alpha \sin \epsilon)^2 + (\cos \alpha \cos \epsilon)^2} \quad (5.17)$$

where  $B$  is the TechSat 21 cluster baseline.

The clutter located in each footprint cell has a certain radar cross section  $\bar{\sigma}$  based on the reflective properties of the illuminated terrain, and is the product of the clutter backscattering coefficient  $\omega$  and the physical area of a single cell  $\theta_{\text{cell}}^2$ :

$$\bar{\sigma} = \omega \Delta \theta_{\text{cell}}^2 R^2 \quad (5.18)$$

According to Eqn. 5.18, the clutter cross-section increases as a function of the square of the range to the target as the grazing angle decreases from nadir to the horizon. Because the clutter cross-section is generally orders of magnitude larger than the RCS of most targets in a space-based GMTI radar, clutter suppression techniques are required during signal processing to enable detection of the target echoes.

As explained earlier, the GINA TechSat 21 radar module uses deconvolution signal processing as it allows for the elimination of clutter in the received radar signals to yield a strictly noise-limited probability of detection [Hacker et al, 2000]. The result of the deconvolution processing is a three-dimensional matrix that contains the spatial location in range and cross-range of each target in the footprint as well as their individual Doppler shifts. Because the received signals are binned in range, the cross-range and Doppler parameters are evaluated for each range bin. Each data point in the matrix contains a target RCS estimate. The target SNR may now be calculated according to Eqn. 5.19 and represents the value one can expect for each cross-range cell in the range bin.

$$\text{SNR} = \frac{P_{\text{avg}} \lambda^2 G_{\text{ap}} G_{\text{af}} \sigma_T \zeta}{(4\pi)^3 R^4 L_s k T_s B W_n} \quad (5.19)$$

where  $P_{\text{avg}}$  is the average transmission power,  $G_{ap}$  is the gain of a single TechSat 21 satellite antenna,  $G_{af}$  is the array factor gain,  $\sigma_T$  is the target radar cross section,  $\zeta$  is the pulse compression ratio,  $L_s$  is the system loss,  $k$  is Boltzmann's constant,  $T_s$  is the system noise temperature, and  $BW_n$  is noise bandwidth. Assuming the received signals are range unambiguous, this computation is repeated for each range bin in the footprint. Eqn. 5.19 represents the SNR calculation over one coherent integration interval for a single aperture. The SNR of a TechSat 21 cluster increases proportionally to the square of the total number of apertures  $N$  in the cluster.

$$\text{SNR} = \frac{P_t \sigma_T A_e \tau \zeta_{\min} N^2}{4\pi\lambda^2 R^4 k L_s T_s} \frac{N^2}{4} n_i^{2/3} n_c \quad (5.20)$$

where  $P_t$  is the transmission power of each satellite,  $A_e$  is the effective antenna area,  $t$  is the pulse width,  $n_i$  is the number of incoherent pulses and  $n_c$  is the number of coherent pulses. Space-based radar systems typically require a minimum SNR of 12-15 dB to ensure the detection of targets in the presence of noise [Cantafio, 1989].

The radar integrity capability metric is the probability of detection  $P_d$ , which is a statistical parameter based on the joint statistics of the noise and target signals. Solving the following equation for  $P_d$  yields the probability of detection after pulse integration:

$$\log\left(\frac{1}{P_d}\right) = \left(\frac{\log(n_{fa})}{n_i^{2/3} n_c \text{SNR}}\right)^{1/\beta} \quad (5.21)$$

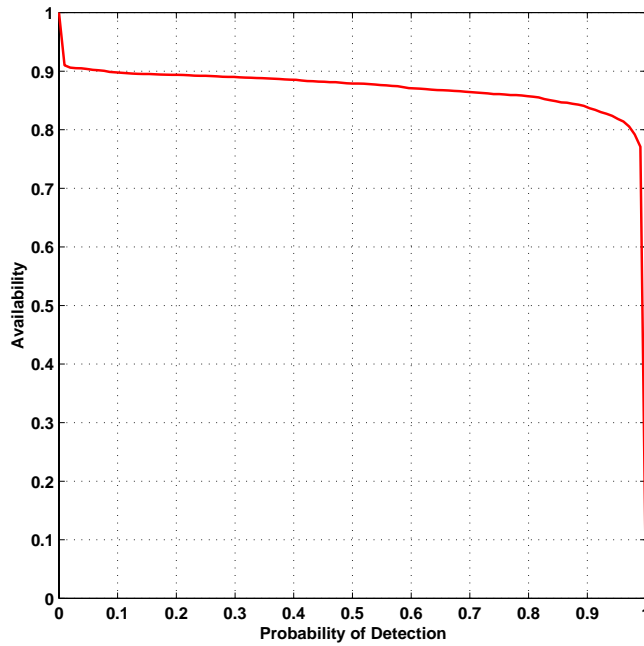
where  $n_{fa}$  is the number of false alarms.  $\beta$  is also a function of the number of incoherent pulses:

$$\beta = \frac{1}{6} + e^{\frac{n_i}{3}} \quad (5.22)$$

and  $n_{fa}$  is computed as a function of the false alarm probability  $P_{fa}$ :

$$n_{fa} = \frac{\ln(0.5)}{\ln(1 - P_{fa})} \quad (5.23)$$

Figure 5.14 illustrates how the probability of detection varies as a function of the availability of the system.



**Figure 5.14** Sample Probability of Detection vs. Availability for a TechSat 21 Cluster [Hacker et al, 2000]

The isolation capability of a space-based GMTI radar system may be characterized by the four metrics of range resolution, cross-range resolution, velocity resolution, and minimum detection velocity. The compressed pulse width  $\tau_c$  drives the range resolution  $\Delta R$ , which is poorest at nadir (i.e.  $\psi=90^\circ$ ):

$$\Delta R = \frac{c\tau_c}{2} \sec \psi \quad (5.24)$$

where  $c$  is the speed of light. Conversely, the maximum baseline between satellites within a single cluster determines the cross-range resolution  $\Delta CR$ , which is best at nadir:

$$\Delta CR = \frac{2\lambda}{B} \frac{R}{\cos \epsilon} \quad (5.25)$$

The amount of time during which coherence is maintained limits the achievable frequency resolution  $\Delta f$  of a TechSat 21 cluster, which in turn drives the velocity resolution of the system  $\Delta v$ :

$$\Delta v = \frac{\lambda}{2} \Delta f \quad (5.26)$$

Finally, the minimum detection velocity (MDV) is derived from the mainlobe clutter return based on the average value of the projected cluster baseline throughout the course of an orbit:

$$\text{MDV} = \frac{\lambda}{2} \left( \frac{4v_p}{B} \sqrt{(\sin \alpha \sin \epsilon)^2 + (\cos \alpha \cos \epsilon)^2} \right) \quad (5.27)$$

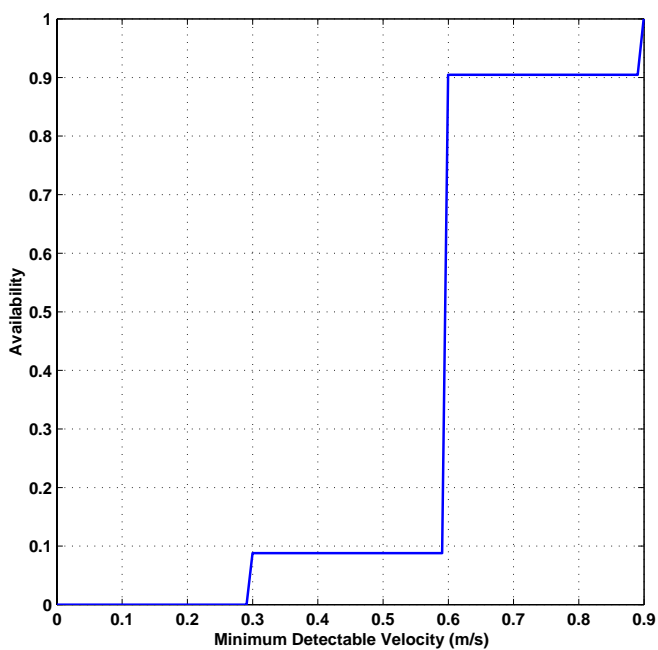
Figure 5.15a illustrates how MDV varies as a function of system availability.

Notice that many of the radar module equations (Eqn. 5.6 through Eqn. 5.27) depend upon the azimuth and elevation angles between each TechSat 21 cluster and the target. These angles, however, are in a continual state of flux as each cluster orbits the Earth. For this reason, probability density functions computed in the constellation module are used in place of discrete numbers for these angles in the radar module. Figure 5.15 illustrates sample plots of the radar isolation and integrity metrics using these probability density functions for the elevation and azimuth angles. For more details on multiaperture interferometry, the details of the TechSat 21 radar module, and the actual code behind the radar module, please refer to the reference by Hacker et al [Hacker et al, 2000].

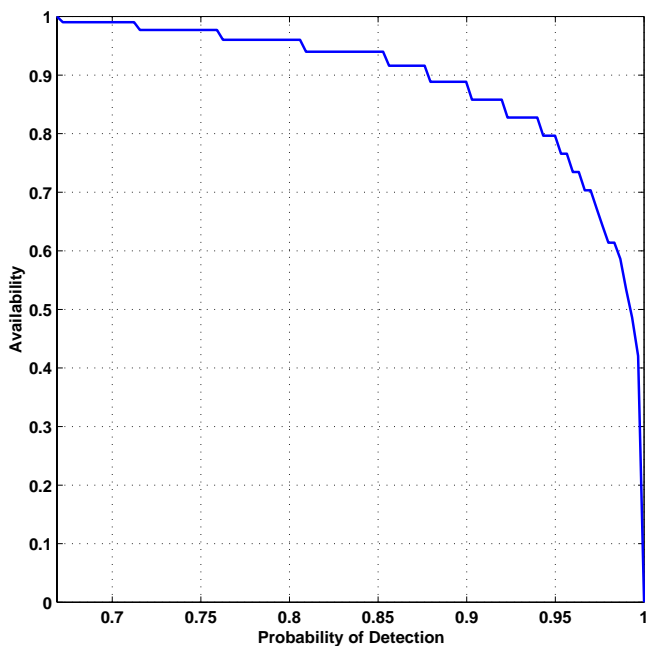
### **Payload Module**

Figure 5.16 illustrates the key inputs and outputs of the payload module. With knowledge of the antenna aperture diameter and radar transmission power from the design vector in combination with the antenna and electronics areal mass and cost densities from the constants vector, the payload, consisting of both the antenna and back-end electronics, is sized





(a)



(b)

**Figure 5.15** Radar Isolation Capability (MDV) (a) and Radar Performance (P(d)) (b) as a Function of Availability Within the System Coverage Area for a Sample TechSat 21 Architecture [Hacker et al, 2000]

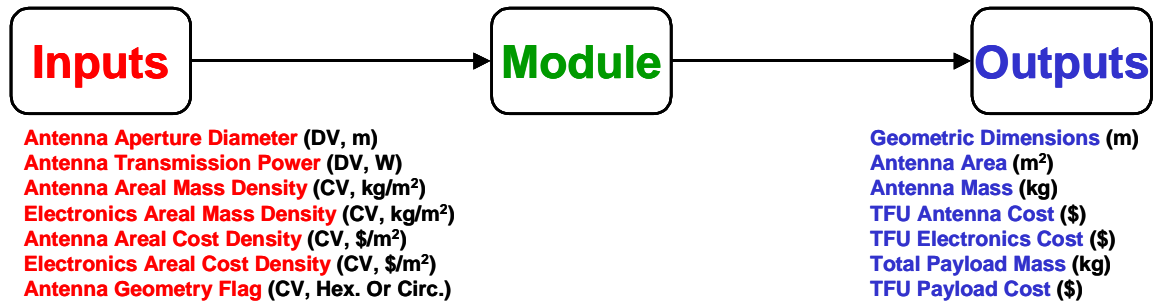


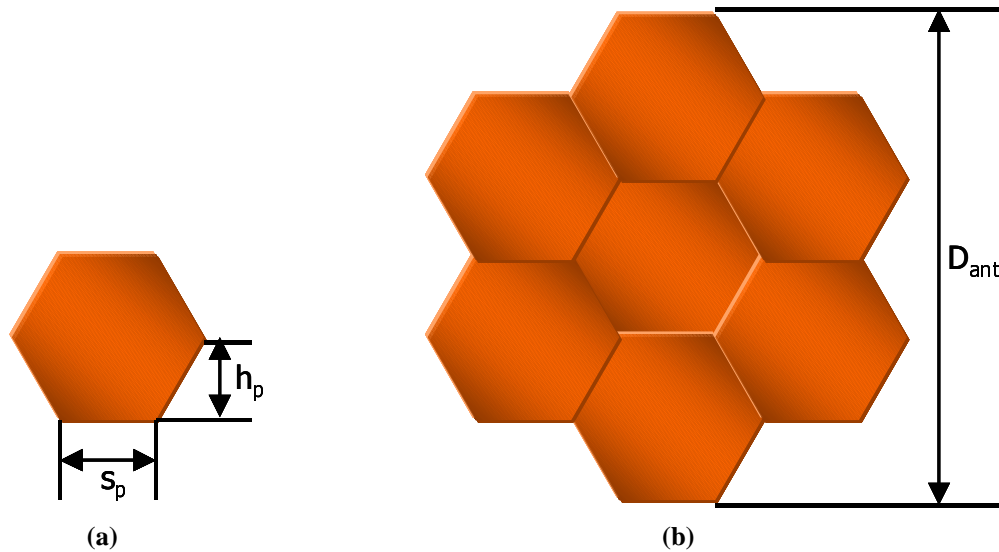
Figure 5.16 Payload Module Inputs and Outputs

to support the radar GMTI mission. Key outputs include the geometric dimensions of the antenna, the total payload mass, and the theoretical first unit (TFU) payload cost.

The TechSat 21 satellite payload is named the transmit/receive antenna module (TRAM). Each TRAM contains a two-dimensional modular phased-array antenna sub-array, with all radio frequency components mounted directly on the exposed antenna face [Jonas et al, 1999]. The AFRL is developing the TRAM to be lightweight, power efficient, and affordable as an enabling technology for future space-based radar systems. The TRAM areal density goal of 6 kg/m<sup>2</sup> represents a 75% reduction over current state-of-the-art space-based phased array antennas [Jonas et al, 1999]. In the TechSat 21 payload module, the TRAM areal density may be set to any desired value, enabling the designer to evaluate the systems wide implications that will result if the 6 kg/m<sup>2</sup> target is not achieved by the AFRL.

A tessellation of seven adjacent hexagonal phased-array antenna panels makes up the TRAM module. Figure 5.17 illustrates the dimensioning nomenclature used by the TechSat 21 payload module to size a single panel (Figure 5.17a) and the entire TRAM module (Figure 5.17b).

With knowledge of the antenna diameter from the design vector, the dimensions of a single hexagonal phased-array panel  $h_p$  and  $s_p$  as well as the total antenna surface area  $A_{\text{ant}}$  may be computed via the following sequence of equations:



**Figure 5.17** TechSat 21 TRAM Module Dimensions of a Single Panel (a) and the Entire Antenna (b)

$$\begin{aligned}
 h_p &= \frac{D_{ant}}{6} \\
 s_p &= \frac{2h_p}{\sqrt{3}} \\
 A_{ant} &= 21s_p h_p
 \end{aligned}
 \tag{5.28}$$

The dimensions of a TechSat 21 satellite in its stowed configuration are identical to the dimensions of a single hexagonal phased-array panel. Thus,  $h_p$  and  $s_p$  are used to determine which launch vehicle payload fairings will and will not support the TechSat 21 satellites as a function of total aperture diameter.

Once the total antenna surface area has been calculated, the mass of the antenna  $m_a$ , mass of the back-end electronics  $m_e$ , and mass of the entire payload  $m_p$  may be computed via the following relationships:

$$\begin{aligned}
 m_a &= \rho_{ma} A_{ant} \\
 m_e &= \rho_{me} A_{ant} \\
 m_p &= m_a + m_e
 \end{aligned}
 \tag{5.29}$$

where  $\rho_{ma}$  and  $\rho_{me}$  are the areal mass densities of the antenna and back-end electronics as set in the constants vector. The TFU payload cost  $C_p$  may also be computed in a similar manner:

$$\begin{aligned} C_a &= \rho_{ca} A_{\text{ant}} \\ C_e &= \rho_{ce} A_{\text{ant}} \\ C_p &= C_a + C_e \end{aligned} \quad (5.30)$$

where  $\rho_{ca}$  is the areal cost density of the antenna,  $\rho_{ce}$  is the areal cost density of the back-end electronics,  $C_a$  is the TFU cost of a single antenna, and  $C_e$  is the TFU cost of a single set of payload back-end electronics.

Once the TFU payload cost is known, the lifecycle payload cost  $C_{PL}$  may be computed as a function of the total number of payloads that will be produced, taking into account the learning curve.  $C_{PL}$  may be computed as:

$$\begin{aligned} \beta_{PL} &= 1 - \frac{\ln\left(\frac{100\%}{S_{PL}}\right)}{\ln 2} \\ L_{PL} &= n_{PL}^{\beta_{PL}} \\ C_{PL} &= C_p L_{PL} \end{aligned} \quad (5.31)$$

where  $S_{PL}$  is the learning curve slope in percent for the number of payloads being produced,  $L_{PL}$  is the learning curve factor, and  $n_{PL}$  is the total number of payloads required for the entire mission (i.e. the sum of the number of initial satellites from the design vector and replenishment satellites from the Markov model).

Thus, the aperture diameter selected in the design vector impacts the lifecycle cost of the TechSat 21 system both directly through TFU payload cost and indirectly through the payload dimensions and mass, which in turn drive launch vehicle selection and associated costs. The TechSat 21 GINA model enables the systems engineer to capture these higher-order couplings and their impact on the system architecture.

## Spacecraft Bus Module

Figure 5.18 illustrates the key inputs and outputs of the spacecraft bus module. With knowledge of the radar antenna transmission power from the design vector, the subsystem component specifications from the constants vector, and the payload parameters from the payload module; each spacecraft subsystem is sized and integrated to create a satellite bus capable of supporting the desired GMTI payload. Key outputs from this module include the mass and power budgets for the following subsystems:

- Propulsion
- Attitude Determination & Control
- Tracking, Telemetry, & Control
- Payload Processing/Command & Data Handling
- Thermal Control
- Power
- Structure

These outputs in turn enable the computation of the total satellite mass and power budgets, which are used downstream within the TechSat 21 GINA model for launch vehicle selection and satellite costing.

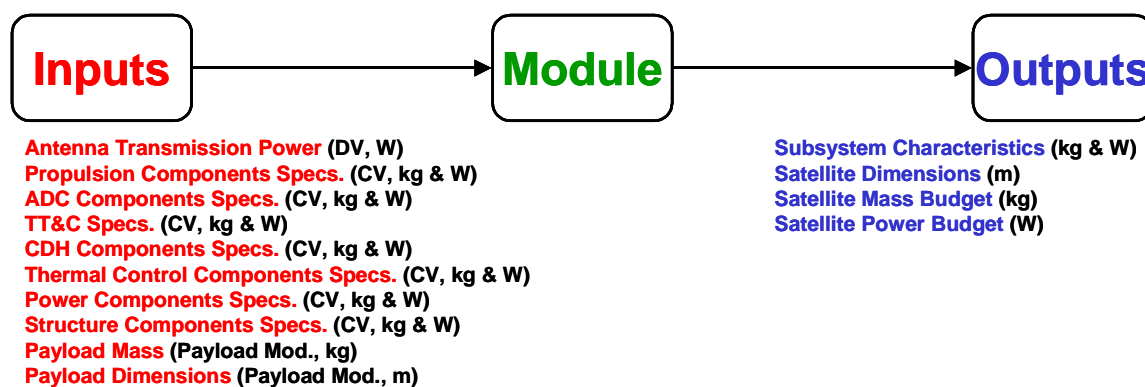
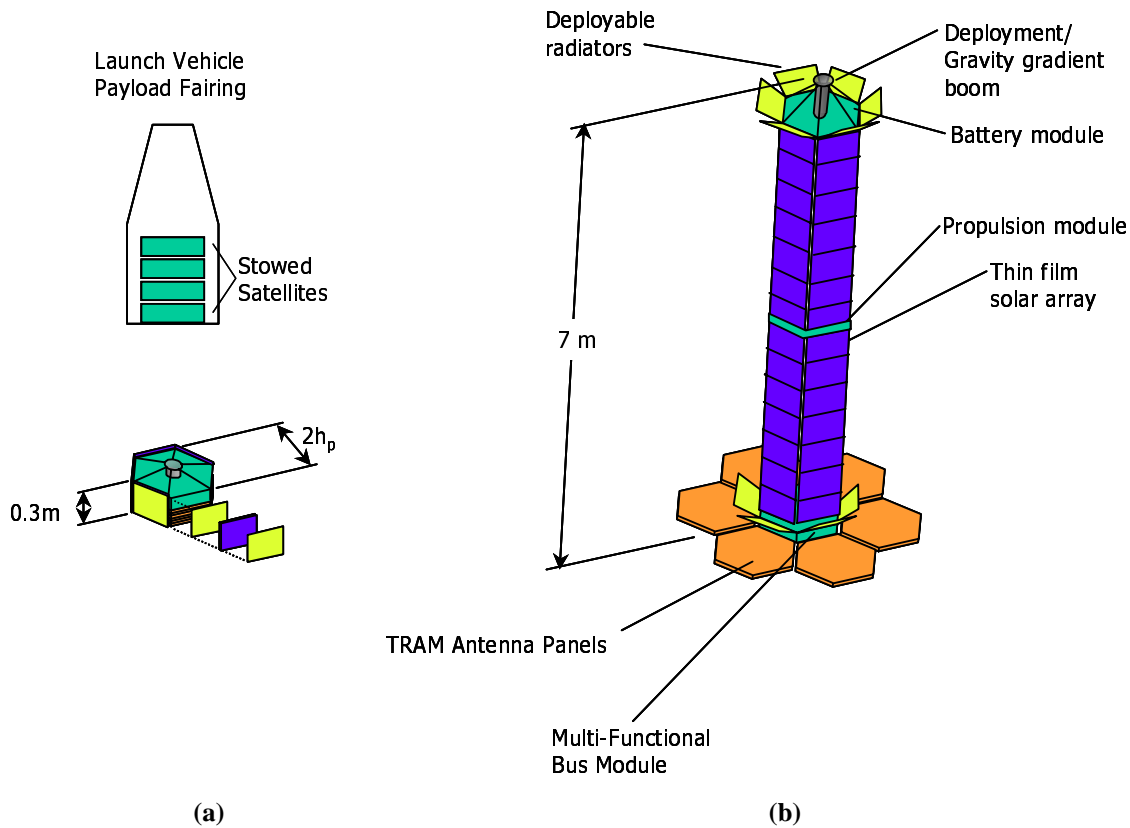


Figure 5.18 Spacecraft Bus Module Inputs and Outputs

The Air Force Research Laboratory has created a preliminary design for the TechSat 21 satellites. Figure 5.19 illustrates this preliminary design in both the stowed (Figure 5.19a)



**Figure 5.19** TechSat 21 Satellite in Stowed (a) and Deployed (b) Configurations [Martin, 2000]

and deployed (Figure 5.19b) configurations. The spacecraft bus module scales this basic AFRL bus design to support the GMTI payload as a function of the radar antenna (i.e. TRAM module) diameter and transmission power.

The propulsion subsystem uses four pulse plasma thrusters for stationkeeping, maneuvers, drag makeup, and deorbit along with ten micro-pulse plasma thrusters for attitude control and relative cluster stationkeeping. The attitude determination and control subsystem employs a combination of star trackers, sun sensors, and magnetic torque rods for satellite attitude knowledge and attitude control. The tracking, telemetry, & control subsystem utilizes the desired number of GEO crosslink, intracluster link, and S-band downlink antennas to maintain communication between satellites within a single cluster and between each cluster and the ground. Multiple radiation-hardened digital signal processors capable of supporting the anticipated 25 GFLOP throughput and 63 Mb memory chips comprise

the payload processing/command & data handling subsystem. Thermal control of the TechSat 21 satellites is achieved with electrochromic surface finishes, multi-layer insulation blankets, and small heaters. The power subsystem uses solar arrays (i.e. MJ GaAs/Ge1, Thin-Film CIS, or MJ Thin-Film), batteries (i.e. Nickel Hydrogen, Lithium Ion, or Lithium Polymer Electrolyte), and a 28 volt bus for satellite power generation, storage, and regulation. Finally, a deployable gravity gradient boom and multifunctional panels create the backbone of the satellite structure.

The CER's in Table 3.4 and Table 3.5 predict the theoretical first unit cost  $TFU_{SC}$  of a TechSat 21 satellite bus. To take into account the fact that multiple TechSat 21 satellite buses will be built, a learning curve is applied to the TFU bus costs. Using the learning curve, the total production cost  $C_{SC}$  for  $n$  TechSat 21 satellite buses is:

$$\beta_{SC} = 1 - \frac{\ln\left(\frac{100\%}{S_{SC}}\right)}{\ln 2} \quad (5.32)$$

$$L_{SC} = n^{\beta_{sc}}$$

$$C_{SC} = TFU_{SC}L_{SC}$$

where  $S_{SC}$  is the learning curve slope in percent and  $L_{SC}$  is the learning curve factor.

### Launch and Operations Module

The launch vehicle and operations module develops and costs an initial constellation deployment strategy, constellation replenishment strategy, and constellation operations strategy. Figure 5.20 illustrates key inputs and outputs of the initial constellation deployment portion of the launch module. This module uses an optimization algorithm to select a subset of launch vehicles from the full set of available launch vehicles under consideration that minimizes total launch costs as a function of the satellite parameters (i.e. mass, volume, number per cluster), orbital parameters (i.e. altitude, inclination, number of orbital planes), and launch vehicle parameters (i.e. performance, cost, fairing dimensions). Key outputs from this module include the maximum number of TechSat 21 satellites each

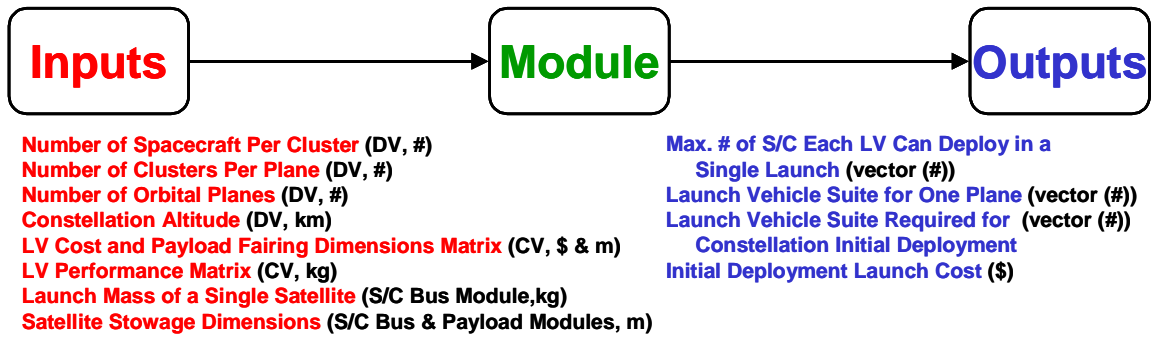


Figure 5.20 Launch Module (Initial Deployment) Inputs and Outputs

available launch vehicle can deploy in a single launch and the recommended suite of launch vehicles for constellation initial deployment.

The AFRL has designated the Pegasus XL, Taurus, Athena 2, and Athena 3 as the target launch vehicles for the deployment of the TechSat 21 constellation. Table 5.5 and Table 5.6 list the constants vector values in the launch vehicle cost and payload fairing dimensions matrix and the launch vehicle performance matrix. The optimization algorithm uses the values in these two matrices to determine the capacity  $K_i$  (i.e. number of TechSat 21 satellites that can be deployed) of each launch vehicle  $i$  to the desired orbit, where the launch vehicle capacity is the lower value of the mass-limited capacity and volume-limited capacity.

TABLE 5.5 Launch Vehicle Cost and Payload Fairing Dimensions Matrix [Isakowitz, 1995]

Launch Vehicle	Cost (\$M)	Fairing Diameter (m)	Fairing Height (m)
Pegasus XL	15	1.2	4.4
Taurus	22	1.4	3.5
Athena 2	27	3.0	9.2
Athena 3	30	3.5	11.2

The integer programming approach presented in Appendix B is used to solve the TechSat 21 constellation launch vehicle selection problem as an optimization problem. The decision variables  $x_i$  in the integer programming formulation of the launch vehicle selection



**TABLE 5.6** Launch Vehicle Performance Matrix (Polar Orbits, inclination~90°) [Isakowitz, 1995]

Launch Vehicle	500 km	600 km	700 km	800 km	900 km	1000 km	1100 km	1200 km	1300 km	1400 km	1500 km
Pegasus XL	280	260	240	220	190	170	150	130	110	80	50
Taurus	900	850	800	750	710	690	650	620	580	550	510
Athena 2	1200	1150	1100	1050	1000	950	900	850	800	750	700
Athena 3	2250	2140	2050	1980	1900	1820	1750	1700	1640	1600	1540

problem represent how many of each launch vehicle  $i$  in Table 5.5 should be used to initially deploy the constellation. The objective function is to select a suite of launch vehicles to deploy the satellite constellation at minimum cost:

$$\text{Min } \sum_{i=1}^n C_i x_i \tag{5.33}$$

where  $C_i$  is the cost of using launch vehicle  $i$ .

Figure 5.21 illustrates the key inputs and outputs of the constellation replenishment portion of the launch module. Notice that the inputs are identical to those of the initial constellation deployment portion of the launch module, with the only difference being the additional inputs related to the Markov model, which estimates the degradation of the constellation as individual satellites fail and need to be replenished over time. In fact, the same integer programming optimization model presented in Appendix B for initial deployment is used, with the only difference being the value of  $N$  in the total satellite deployment constraint. This new value for the number of TechSat 21 satellites that need to be replenished by subsequent launches is computed via the Markov model.

Markov reliability modeling techniques are used to determine both the probability that the system will continue to function over a given amount of time and the likelihood with which the system will function in different partially failed states throughout the mission [Jilla & Miller, 1997]. From the Markov model, a system of differential equations is developed, as shown in Eqn. 5.34, where  $\dot{\vec{P}}$  is the time rate of change of the state probabil-

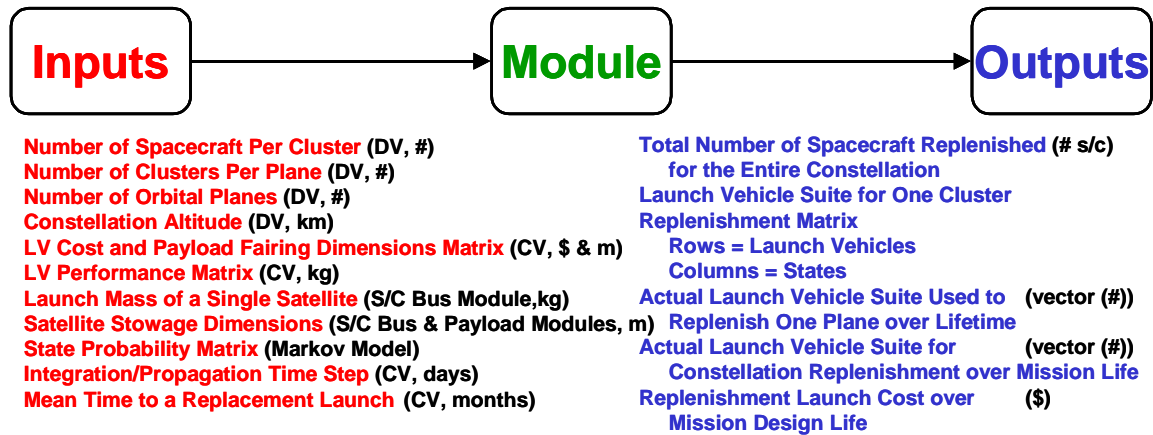


Figure 5.21 Launch Module (Replenishment) Inputs and Outputs

ity vector,  $A$  is the state coefficient matrix comprised of component failure rates, and  $\vec{P}$  is the state probability vector.

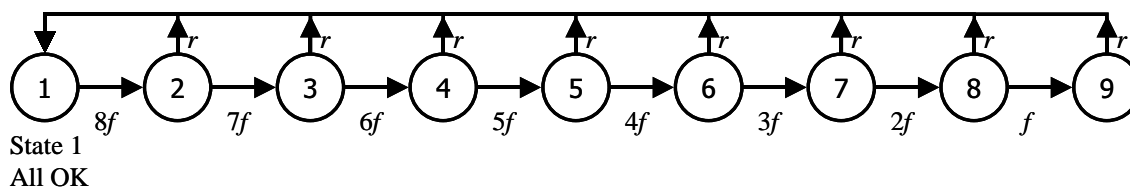
$$\dot{\vec{P}} = A\vec{P} \quad (5.34)$$

Take the example of an eight satellite per cluster TechSat 21 architecture. To apply a Markov modeling methodology, the states of the system must be time dependent, sequential, and mutually exclusive. If the system satisfies these requirements, then a set of differential equations can be written to model the evolution of the system by taking advantage of the Markov property, which states that given full knowledge of the current state of the system, one can predict all future states by integrating a set of differential equations, irrespective of the past states of the system [Gelb, 1974].

To develop the proper set of differential equations for this system, a Markov model state diagram illustrating each possible state of the system must first be created for each TechSat 21 architecture. Figure 5.22 illustrates the Markov model state diagram for a single eight-satellite TechSat 21 cluster. Because a minimum of two satellites are required for IFSAR, this model contains seven possible functioning states:

- State 1 – All eight satellites are working,
- State 2 – Seven of the eight satellites are working (i.e. 1 failed),
- State 3 – Six of the eight satellites are working (i.e. 2 failed),

- State 4 – Five of the eight satellites are working (i.e. 3 failed),
- State 5 – Four of the eight satellites are working (i.e. 4 failed),
- State 6 – Three of the eight satellites are working (i.e. 5 failed), and
- State 7 – Two of the eight satellites are working (i.e. 6 failed).



**Figure 5.22** Aggregated Markov Model State Diagram for an Eight-Satellite TechSat 21 Cluster

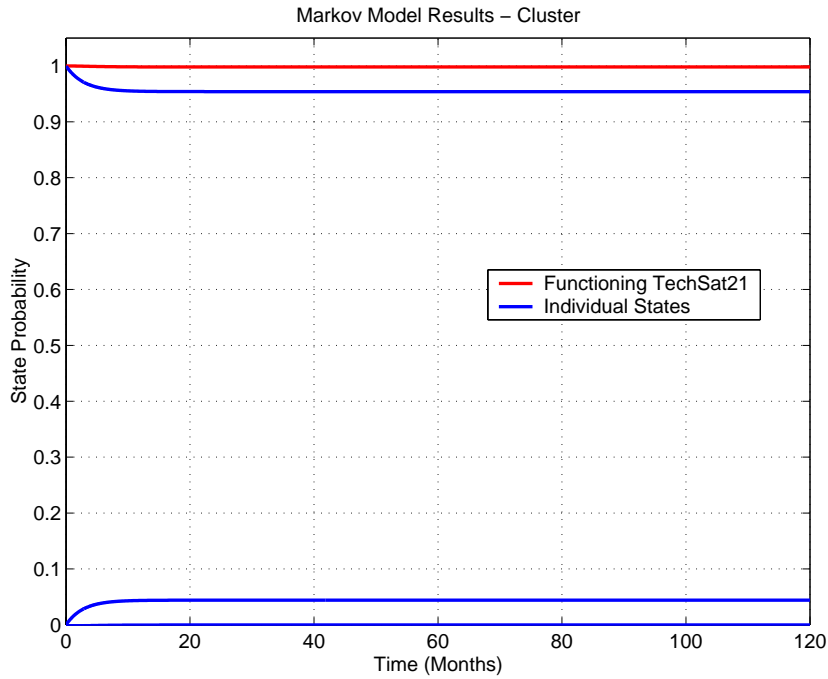
The remaining two states in the Markov model state diagram (i.e. state 8 (one of the eight satellites is working (seven have failed)) and state 9 (all eight satellites have failed)) represent system failure states as IFSAR may no longer be performed. It should also be noted that even though IFSAR may be performed in states 1-6, the system requirements on minimum detection velocity, probability of detection, etc. may or may not be met depending upon the other parameters of the TechSat 21 cluster (i.e. aperture diameter, transmission power, etc.).

From the Markov model state diagram, a system of differential equations can be written to determine the probability of the system being in any given state at any given time. This is done by representing each possible state of the system in the Markov model as a node in a network (Figure 5.22). To determine the differential equation for a particular state, the flow in and out of the node representing that state is balanced. The eight-satellite per cluster TechSat 21 architecture requires a set of nine partially coupled, linear, first-order differential equations to model the system (Eqn. 5.35).

$$\begin{bmatrix} \dot{P}_1 \\ \dot{P}_2 \\ \dot{P}_3 \\ \dot{P}_4 \\ \dot{P}_5 \\ \dot{P}_6 \\ \dot{P}_7 \\ \dot{P}_8 \\ \dot{P}_9 \end{bmatrix} = \begin{bmatrix} -8f & r & r & r & r & r & r & r & r \\ 8f & -(r+7f) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7f & -(r+6f) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6f & -(r+5f) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5f & -(r+4f) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4f & -(r+3f) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3f & -(r+2f) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2f & -(r+f) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & f & -r \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \\ P_7 \\ P_8 \\ P_9 \end{bmatrix} \quad (5.35)$$

Knowledge of the failure rates  $f$  and replenishment rates  $r$  for the TechSat 21 satellites are required to solve for the state probabilities. The system's initial conditions are also required for the solution. In every case, the initial conditions (i.e. at time=0 in the beginning of the operational mission) are a 100% probability of being in state one and a 0% probability of being in all successive states.

The plot in Figure 5.23 illustrates the results for the eight-satellite per cluster example. The Markov model gives the exact probability of being in any of the seven operational states as a function of time throughout the ten year (i.e. 120 month) TechSat 21 mission. The red line in the plot represents the sum probability of being in any of the seven operational states. Each blue line represents a unique operational state. For example, the top blue line denotes state 1 in which all eight satellites are functioning. Notice that as the satellite failure and replenishment rates are constant, a probabilistic equilibrium is reached for each possible state of the system. In this particular case, with the assumed failure and replenishment rates, the steady-state probability is 99.8% that two or more satellites in a single TechSat 21 cluster are functional, 95.4% that all eight satellites are functional, 4.4% that exactly seven out of eight satellites are functional, and so forth. Markov models for TechSat 21 architectures with 4 to 16 satellites per cluster were implemented in the same manner.



**Figure 5.23** Markov Model Results for the Eight-Satellite TechSat 21 Cluster ( $f=0.0002 \text{ month}^{-1}$ ,  $r=0.33 \text{ month}^{-1}$ )

The solution to the system of differential equations in Eqn. 5.35 determines the probability of the system being in any given state at a particular time. The total number of satellites  $Y_{sc}$  that require replenishment may now be calculated:

$$Y_{sc} = n_{cl} \left( \int_1^{MDL} \sum_{i=1}^n S_{ri} P_i(t) dt \right) \tag{5.36}$$

where the limits of integration denote time steps of one day,  $n_{cl}$  is the total number of clusters in the constellation,  $MDL$  is the mission design life (i.e. the total number of days in the mission),  $i$  is an index indicating a particular system state (i.e. when  $i=1$  all satellites are working; when  $i=2$  one satellite has failed, etc.),  $n$  is the total number of system states ( $n=9$  for the eight satellites per cluster example),  $S_{ri}$  is the number of satellites that need to be replaced within a cluster when the system is in state  $i$ , and  $P_i(t)$  is the probability of being in any state  $i$  as a function of time  $t$  from Eqn. 5.34.

Likewise, the total number of launch vehicles  $\vec{Y}_{LV_r}$  required for replenishment is:

$$\vec{Y}_{LV_r} = n_{op} \left( \int_1^{MDL} \sum_{i=1}^n \vec{L}_{r_i} P_i(t) dt \right) \quad (5.37)$$

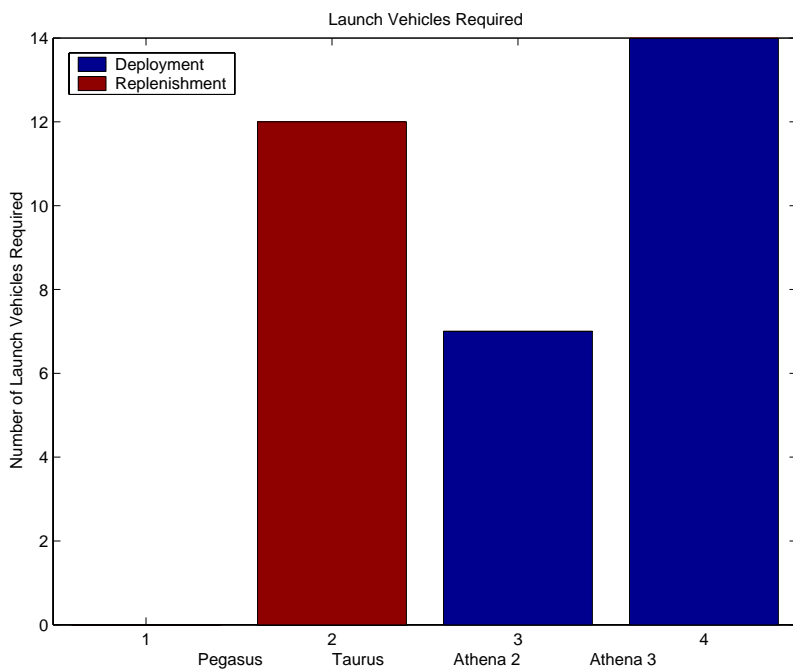
where  $n_{op}$  is the total number of orbital planes in the constellation and  $\vec{L}_{r_i}$  is a vector of the number of each launch vehicle (i.e. Pegasus, Taurus, Athena 2, and Athena 3) required to completely replenish an orbital plane when the system is in state  $i$ , which is computed by solving the integer program in Appendix B for each state  $i$  with  $N$  in the total satellite deployment constraint set to the value of the product of  $S_{ri}$  with the total number of clusters per orbital plane and  $OP$  in the orbital dynamics constraint set to a value of one.

Finally, the lifecycle launch costs  $C_L$  may be computed as the sum of the initial deployment and replenishment costs:

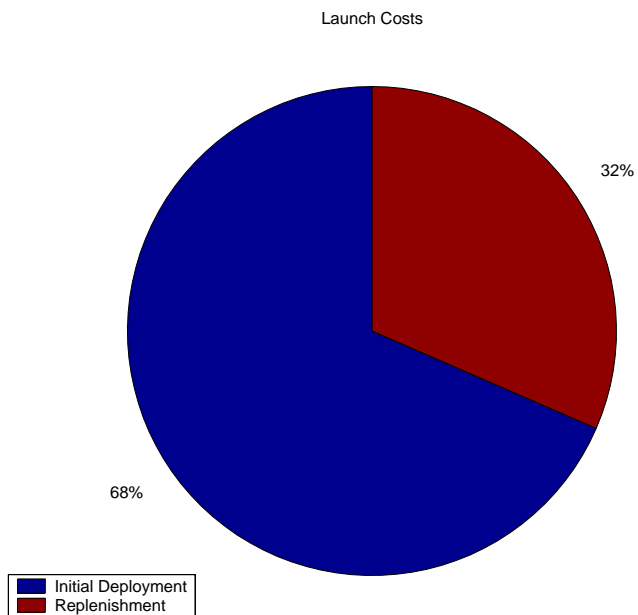
$$C_L = \left( \sum_{i=1}^n C_{LV_i} x_i \right) + \left( \sum_{i=1}^n C_{LV_i} Y_{LV_{r_i}} \right) \quad (5.38)$$

where  $n$  is the total number of launch vehicles being considered,  $C_{LV_i}$  is the cost of launch vehicle  $i$ ,  $x_i$  is the number of each launch vehicle  $i$  selected by the optimization algorithm in Appendix B for initial deployment, and  $Y_{LV_{r_i}}$  is the number of each launch vehicle  $i$  selected by the optimization algorithm in conjunction with the Markov model for constellation replenishment.

Figure 5.24 illustrates sample outputs from the launch initial deployment and replenishment modules for the Aerospace Corporation baseline TechSat 21 designs. Figure 5.24a shows the optimum (i.e. minimum cost) suite of launch vehicles for initial deployment and replenishment, while Figure 5.24b shows the relative cost breakdown of the \$854 M launch budget between initial deployment (\$574 M) and replenishment (\$280 M) over the ten-year mission life.



(a)



(b)

**Figure 5.24** Launch Module Outputs for The Aerospace Corporation TechSat 21 Baseline Design

Figure 5.25 illustrates key inputs and outputs of the operations portion of the launch and operations module. With knowledge of the total number of satellites within the TechSat 21 constellation, operations labor cost, and operations learning curve slopes; the operations crew size and costs and the software size and cost may be estimated. Key outputs include lifecycle operations cost and lifecycle software cost.

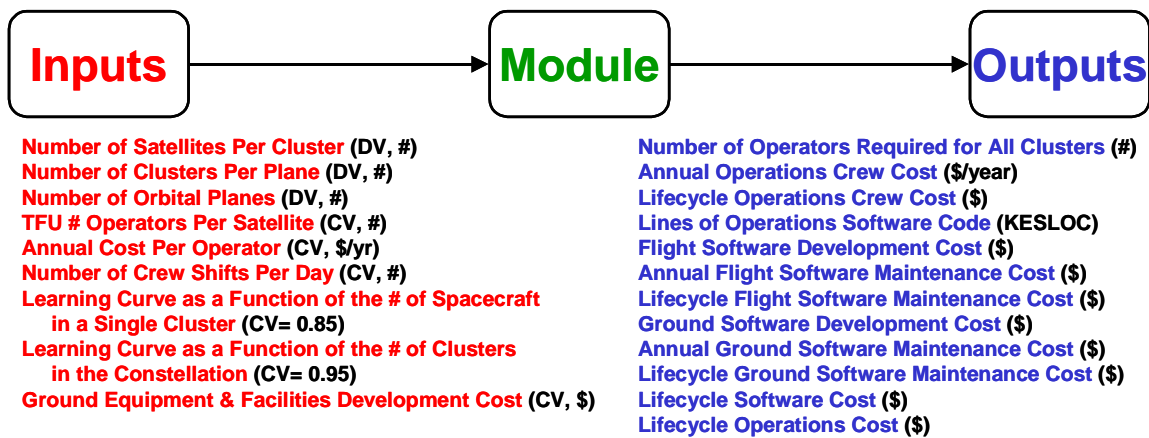


Figure 5.25 Operations Module Inputs and Outputs

The TechSat 21 constellation will likely contain hundreds of satellites. The current methods for operating spacecraft – which include maintaining a team of several orbital analysts, mission operators, and spacecraft engineers per satellite [Boden & Larson, 1996] will not work due to the prohibitive costs that will result from operating hundreds of satellites simultaneously. Rather, newer methods demonstrated with the operation of such systems as the Iridium constellation, which used the capability provided by inter-satellite links to reduce the number of mission operators to one per orbital plane, must be employed to reduce lifecycle operations costs [Rolle, 1999]. The TechSat 21 operations module utilizes an operations learning curve to model the operation of the TechSat 21 constellation.

The TechSat 21 operations module employs two separate learning curves [Larson & Wertz, 1992] to estimate the size of the constellation operations crew. The first learning curve computes the total number of operators (i.e. orbital analysts, mission operators, and



spacecraft engineers)  $n_{ocl}$  that are required to operate a single TechSat 21 cluster as a function of the total number of satellites  $n_{scl}$  in the cluster:

$$\begin{aligned}\beta_{cl} &= 1 - \frac{\ln\left(\frac{100\%}{S_{cl}}\right)}{\ln 2} \\ L_{cl} &= n_{scl}^{\beta_{cl}} \\ n_{ocl} &= n_{osc}L_{cl}\end{aligned}\tag{5.39}$$

where  $S_{cl}$  is the learning curve slope in percent for the number of satellites in a cluster,  $L_{cl}$  is the learning curve factor for the cluster, and  $n_{osc}$  is the estimated number of operators required to support a single TechSat 21 satellite as set in the constants vector.

The second learning curve computes the total number of operators  $n_{oco}$  that are required to operate the entire constellation as a function of the total number of clusters  $n_{cl}$  in the constellation:

$$\begin{aligned}\beta_{co} &= 1 - \frac{\ln\left(\frac{100\%}{S_{co}}\right)}{\ln 2} \\ L_{co} &= n_{cl}^{\beta_{co}} \\ n_{oco} &= n_{ocl}L_{co}\end{aligned}\tag{5.40}$$

where  $S_{co}$  is the learning curve slope in percent for the number of clusters in the constellation,  $L_{co}$  is the learning curve factor for the constellation, and  $n_{ocl}$  is the estimated number of operators required to operate a single TechSat 21 cluster as computed in Eqn. 5.39. With knowledge of  $n_{oco}$ , the annual operations crew cost  $C_{cr}$  may now be calculated as:

$$C_{cr} = s_d n_{oco} C_{em}\tag{5.41}$$

where  $s_d$  is the number of operations control center shifts per day and  $C_{em}$  is the annual cost (i.e. salary and overhead) per employee.

The operations module also estimates the total amount of software required to operate the TechSat 21 constellation along with the associated software costs. The following relationships have been derived based on the information provided by The Aerospace Corporation concerning one of their TechSat 21 point designs. The total amount of flight (KESLOC<sub>f</sub>) and ground (KESLOC<sub>g</sub>) software – measured in units of thousands of estimated software lines of code (KESLOC) – may be estimated as a function of the total number of satellites in a single cluster ( $n_{scl}$ ):

$$\begin{aligned} \text{KESLOC}_f &= 100 + 3.6n_{scl}^2 \\ \text{KESLOC}_g &= 400 + 7.2n_{scl}^2 \end{aligned} \quad (5.42)$$

The flight ( $C_{dfsw}$ ) and ground ( $C_{dgsW}$ ) software development costs are then:

$$\begin{aligned} C_{dfsw} &= (781 \times 10^3)(\text{KESLOC}_f) \\ C_{dgsW} &= (331 \times 10^3)(\text{KESLOC}_g) \end{aligned} \quad (5.43)$$

The annual flight ( $C_{mfsW}$ ) and ground ( $C_{mgsw}$ ) software maintenance costs are:

$$\begin{aligned} C_{mfsW} &= \kappa_f C_{dfsw} \\ C_{mgsw} &= \kappa_g C_{dgsW} \end{aligned} \quad (5.44)$$

where  $\kappa_f$  and  $\kappa_g$  are the annual flight and ground software maintenance cost fractions respectively.

Finally, total lifecycle operations cost  $C_{OPS}$  for the TechSat 21 constellation may be estimated as:

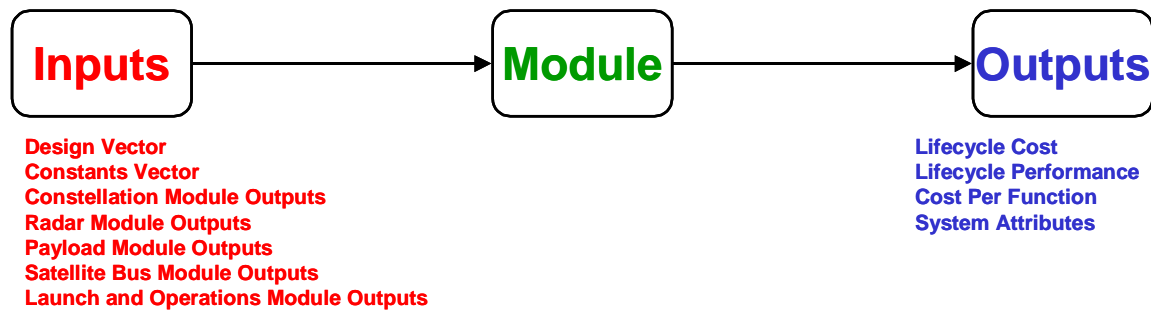
$$C_{OPS} = MDL(C_{cr} + C_{mfsW} + C_{mgsw} + C_{mgf}) + C_{dfsw} + C_{dgsW} + C_{dgf} \quad (5.45)$$

where  $MDL$  is the mission design life,  $C_{mgf}$  is the annual maintenance cost of the ground facilities, and  $C_{dgf}$  is the initial development cost of the ground facilities. It should be noted that this portion (i.e. the operations module) of the TechSat 21 GINA model is subject to high uncertainty because there currently exists no operations cost data to bench-

mark against – there exist no staffed control centers for operating a constellation of satellite clusters. Uncertainty analysis and propagation within the context of GINA and MMDOSA is an area of continuing research [Walton & Hastings, 2001].

**Systems Analysis Module**

Figure 5.26 illustrates the inputs and outputs of the systems analysis module. With knowledge of the outputs from all the preceding modules in the TechSat 21 GINA model, the system attributes that will be used to compare various conceptual design architectures may be computed. Key outputs include lifecycle cost, lifecycle performance, and cost per function.



**Figure 5.26** Systems Analysis Module Inputs and Outputs

The first primary output of the systems analysis module is lifecycle cost. Three primary cost categories are the development of the spacecraft payloads and buses, deployment (i.e. launch) of the system, and operation of the system. Methods for estimating each of these costs have been covered in the previous sections of this chapter.

In the TechSat 21 GINA model, lifecycle cost  $C_{LCC}$  is computed according to the following equation:

$$C_{LCC} = C_{PL} + C_{SC} + C_L + C_{OPS} \tag{5.46}$$

where  $C_{PL}$  is the lifecycle payloads cost (Eqn. 5.31),  $C_{SC}$  is the lifecycle satellite bus cost (Eqn. 5.32),  $C_L$  is the lifecycle launch cost (Eqn. 5.38), and  $C_{OPS}$  is the lifecycle operations cost (Eqn. 5.45).

The second primary output of the systems analysis module is lifecycle performance. The solution to the Markov model system of differential equations (Eqn. 5.34 and Eqn. 5.35) determines the probability of the TechSat 21 system being in any given state at a particular time. Coupling the outputs of the reliability model with the outputs of the radar module yields the total lifecycle performance (i.e. estimated lifecycle probability of detection  $\Psi_{Pd}$ ). The coupling equation is:

$$\Psi_{Pd} = \frac{\int_1^{MDL} \sum_{i=1}^n P_{di} P_i(t) dt}{MDL} \quad (5.47)$$

where the limits of integration denote time steps of one day,  $MDL$  is the mission design life (i.e. the total number of days in the mission),  $i$  is an index indicating a particular operational state (i.e. when  $i=1$  all satellites are working; when  $i=2$  one satellite has failed, etc.),  $n$  is the total number of operational states (i.e.  $n=7$  for the eight-collector aperture example),  $P_{di}$  is the probability of detection of the system in each state  $i$ , and  $P_i(t)$  is the probability of being in any state  $i$  as a function of time  $t$  (i.e. from Eqn. 5.34 for any generic TechSat 21 architecture and Eqn. 5.35 for the eight-satellite per cluster TechSat 21 architecture).

The third primary output of the systems analysis module is the cost per function metric. For the TechSat 21 GMTI mission, the CPF is defined as the cost per unit probability of detection, and is calculated by dividing the total lifecycle cost of a TechSat 21 architecture (Eqn. 5.46) by the total lifecycle system performance (Eqn. 5.47).

$$CPF = \frac{C_{LCC}}{100\Psi_{Pd}} \quad (5.48)$$

### **Benchmarking the TechSat 21 GINA Model**

The TechSat 21 GINA model has been benchmarked against a specific TechSat 21 constellation point design created and documented by The Aerospace Corporation's Conceptual Design Center (CDC) [Stallard, 1999]. The goal of the TechSat 21 benchmarking exercise is not to match exactly the results from the existing point design, but rather to identify coarse correlations, major discrepancies, and the sources of any such major discrepancies; and then to iterate upon the GINA model to improve overall model fidelity. The benchmark point design may be entered manually by the user via the TechSat21 GINA model graphical user interface (GUI) or within the body of the code in the design vector module. Figure 5.27 illustrates the GUI with The Aerospace Corporation CDC point design inputs, and Table 5.7 lists the complete design vector along with several of the key constants vector inputs for the CDC point design.

Figure 5.28 compares the mass of each TechSat 21 satellite subsystem as computed by the TechSat 21 GINA model (TMAS), the Aerospace Corporation Conceptual Design Center (CDC), and by the Air Force for the TechSat 21 flight experiment (Flt. Exp.). Notice that while the outputs of the GINA model fit well with the flight experiment data, some discrepancies do exist with the CDC point design data. The GINA model significantly underestimates the payload mass in comparison with the CDC model. This discrepancy was reported to The Aerospace Corporation, and it was discovered that the GINA model's initial estimates of the areal density for the payload TRAM antenna and backend electronics were too low. The values for these two parameters were then updated in the constants vector to  $6 \text{ kg/m}^2$  and  $5 \text{ kg/m}^2$  respectively to correct for the discrepancy.

Figure 5.29 compares the cost of each component of the TechSat 21 system as computed by version one of the GINA model (TMAS 1), version two of the GINA model (TMAS 2), and the CDC model (AFRL CDC). Notice the improvement in the correlation of the GINA model with the CDC model between versions one and two of the GINA model. This illustrates the importance of benchmarking as a process to improve the fidelity of any DSS GINA model. As was the case in the payload mass calculation, a major discrepancy

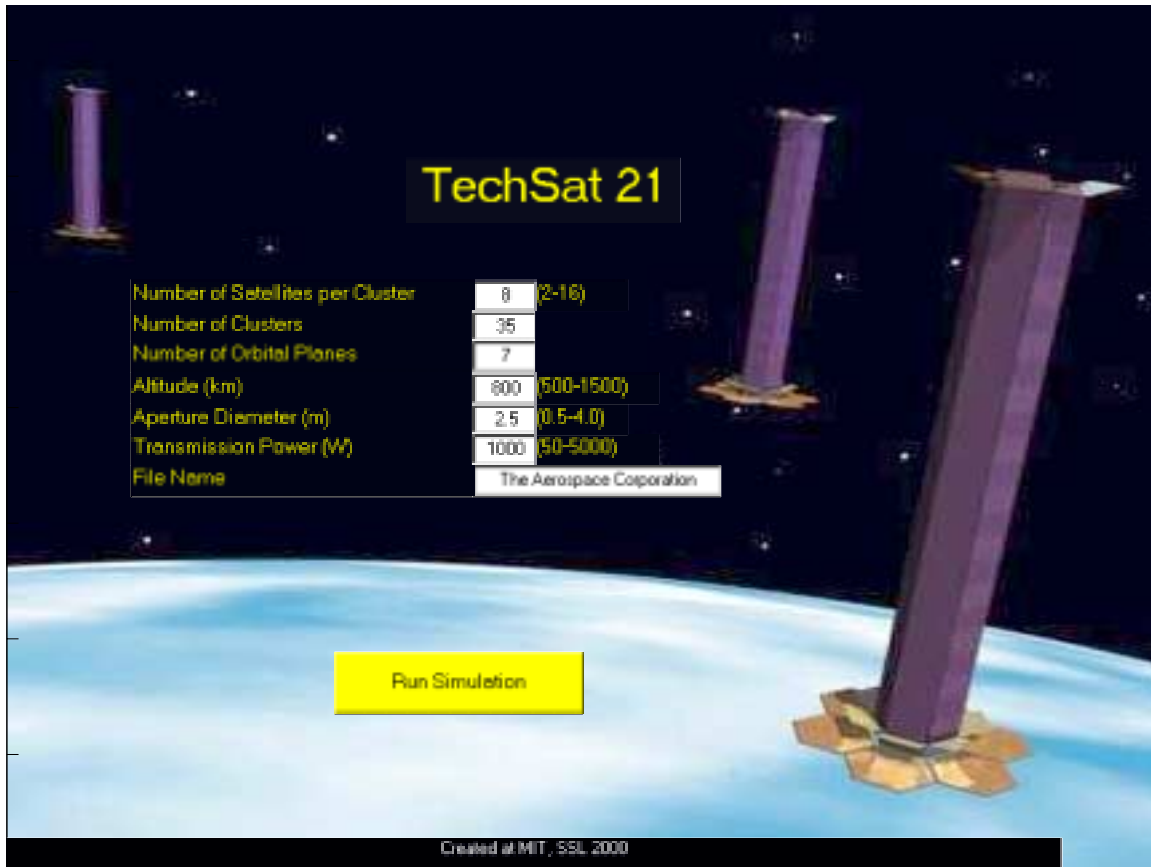


Figure 5.27 TechSat 21 GINA Model Graphical User Interface

TABLE 5.7 Aerospace Corporation CDC Point Design Inputs for TechSat 21 GINA Model Benchmark

Design Vector Parameters	Value	Constants Vector Parameters	Value
# Satellites Per Cluster	8	Mission Design Life	10 years
Aperture Diameter	2.5 m	Plane Phasing	6 PU
Radar Payload Power	1000 W	Max. Cone Half-Angle	58°
Constellation Altitude	800 km	Constellation Type	Polar
# Clusters Per Orbital Plane	5	Radar Frequency	10 GHz
# Orbital Planes	7	Bus Mast Length	7 m

existed between version one of the GINA model and the CDC model in the lifecycle payload cost calculation. Under the guidance of The Aerospace Corporation and the AFRL, the constants vector parameters for the payload TRAM antenna and backend electronics were revised upwards from 250 \$K/m<sup>2</sup> to 1000 \$K/m<sup>2</sup>. This simple change illustrates the

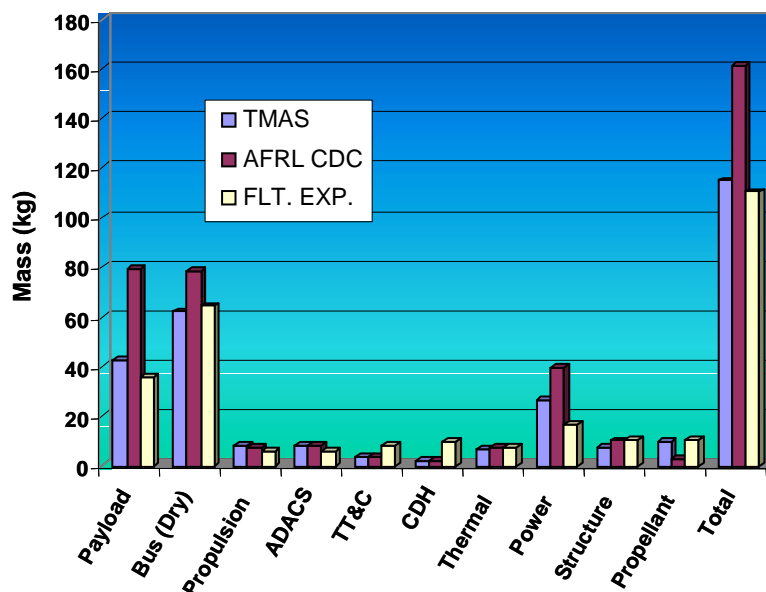


Figure 5.28 Mass Comparisons for the TechSat 21 GINA Model Benchmark

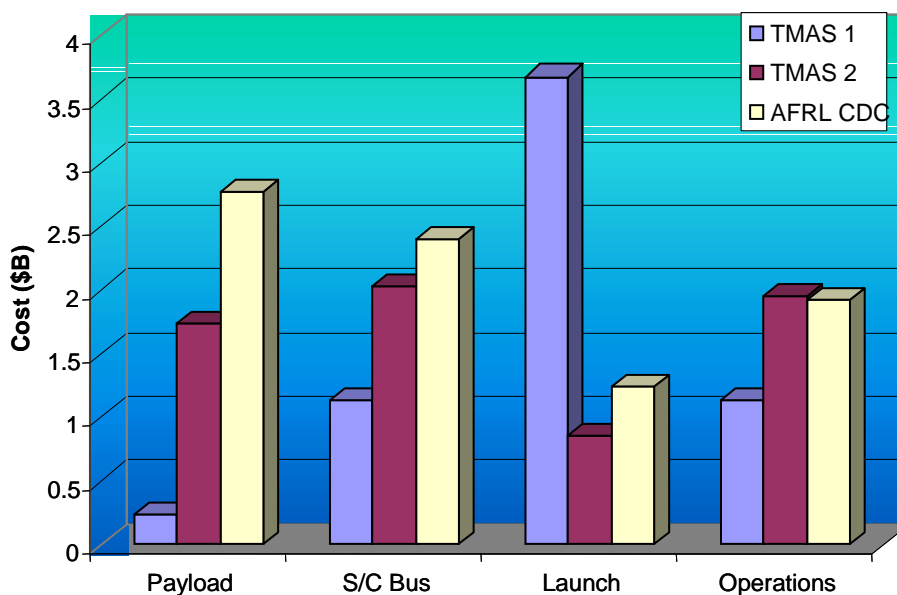


Figure 5.29 Cost Comparisons for the TechSat 21 GINA Model Benchmark

ease with which the any GINA model may be modified due to the inherent modularity of the GINA modeling approach. The correlation of the spacecraft bus lifecycle cost was improved from version one to version two of the GINA model by reducing the number of

CER's used in the calculation to those which were deemed most relevant to the TechSat 21 satellite bus design.

A large discrepancy, greater than \$2B existed in the estimated lifecycle launch cost between version one of the GINA model and the CDC model. This discrepancy occurred due to the fact that in version one of the GINA model, the launch vehicle selection optimization algorithm (Appendix B) was formulated to optimize the deployment of a single cluster. The solution was then multiplied by the total number of clusters in the constellation. This formulation ignored the fact that while it is costly from a fuel perspective to change the orbital plane of a satellite once it has been deployed in a circular low Earth orbit, it is relatively easy and cheap from a fuel perspective to change the phasing of satellites with the same LEO orbital plane. For example, two clusters of satellites may be deployed to the same orbit by a single launch vehicle, and then the satellites could be phased to create two separate clusters within the same orbital plane. Thus, the formulation of the launch vehicle selection optimization algorithm was modified from a single cluster focus to an orbital plane focus in version two of the GINA model. The total launch vehicle suite required to deploy and replenish the TechSat 21 constellation is determined by multiplying the solution to this new optimization problem by the total number of planes in the constellation. In fact, the optimization algorithm now finds a more cost-effective launch vehicle solution than the CDC proposal. Finally, the cost modeling fidelity of the operations module was also improved with additional data from The Aerospace Corporation, the AFRL, and the U.S. Space & Missiles Command on the operations concept for the TechSat 21 constellation. The results from this section illustrate the importance of benchmarking a GINA model and the subsequent improvements in model fidelity that result from an iterative process.

### **5.2.6 GINA Step 6 – Explore the System Trade Space**

To demonstrate the ability of the GINA model to explore the system trade space for TechSat 21, the local trade space around The Aerospace Corporation's CDC point design for



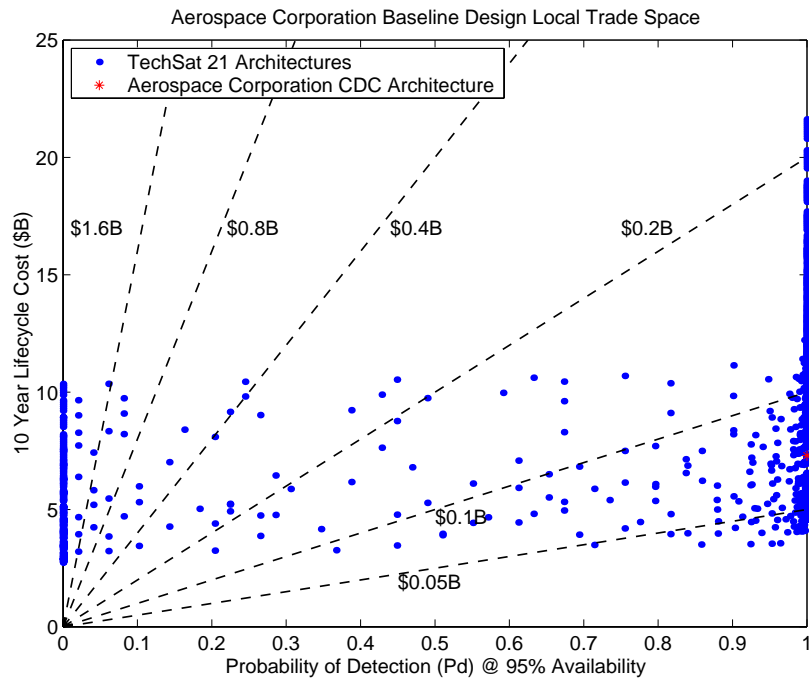
TechSat 21 (Table 5.8) is investigated. Figure 5.30 plots the performance vs. cost of the CDC design (i.e. red asterisk) and 1040 permutations (i.e. blue dots) of the CDC design. In the 1040 permutations, the constellation altitude (i.e. 800 km), number of orbital planes (i.e. 7), and the number of clusters per orbital plane (i.e. 5) are held constant; while the number of satellites per cluster (i.e. 4-16), antenna aperture diameter (i.e. 0.5 m - 4.0 m), and antenna payload power (i.e. 100 W - 1000 W) are varied across all possible combinations. Notice that while the CDC architecture does provide excellent GMTI performance, there appear to exist other architectures within the global trade space that provide similar performance at a significantly reduced cost. Also note in the zoom-in plot that lifecycle cost increases with probability of detection for the most cost-effective system architectures.

**TABLE 5.8** Aerospace Corp. CDC Point Design Inputs for TechSat 21 Local Trade Space Exploration

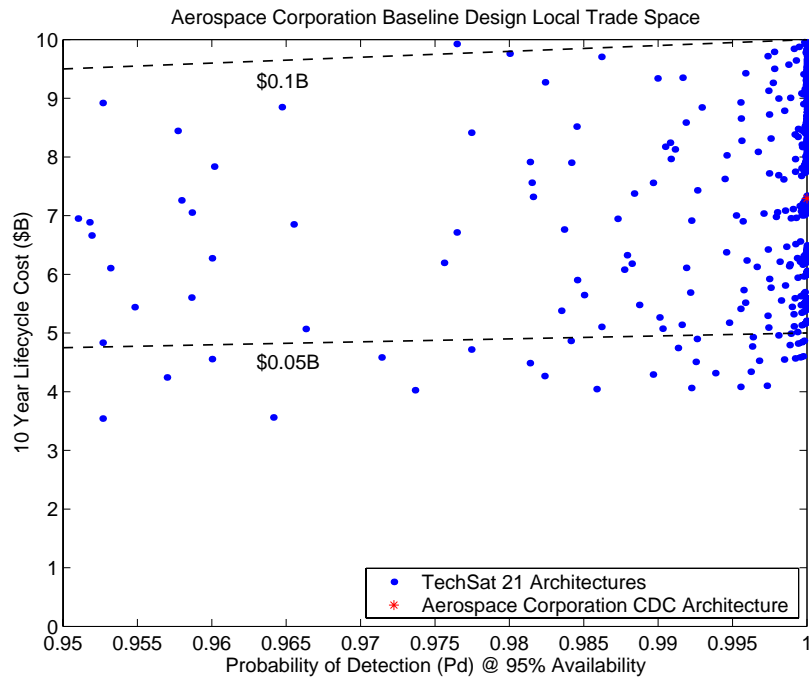
<b>Design Vector Parameters</b>	<b>Value</b>	<b>Constants Vector Parameters</b>	<b>Value</b>
Satellites Per Cluster	4-16	Mission Design Life	10 years
Aperture Diameter	0.5 m - 4.0 m	Plane Phasing	6 PU
Radar Transmission Power	100 W - 1000 W	Max. Cone Half-Angle	58°
Constellation Altitude	800 km	Constellation Type	Polar
# Clusters Per Orbital Plane	5	Radar Frequency	10 GHz
# Orbital Planes	7	Bus Mast Length	7 m

These results do not mean that The Aerospace Corporation CDC architecture is a poor design. However, this example does illustrate how the systems engineer may use a tool like GINA in combination with the MDO algorithms presented in Section 5.5 to find more cost-effective families of system architectures during the conceptual design phase, when the systems engineer has the most flexibility in making design decisions that will determine the lifecycle cost of the system.

At this point, it is important to remember that the results in Figure 5.30 are contingent upon three items:



(a)



(b)

Figure 5.30 Aerospace Corporation CDC Design Trade Space (a) and Trade Space Zoom-In (b)

1. The System Requirements,
2. Technology Assumptions (i.e. solar cell blanket areal density, launch vehicle costs, battery depth of discharge, etc.) in the constants vector, and
3. The mathematical models within each TechSat 21 GINA model module (Figure 5.5).

The real power of a GINA model lies in the ability to change any and all of these three items, and then see how the system trade space changes. For example, by changing the solar cell blanket areal density value in the constants vector, the systems engineer may estimate how the lifecycle performance and cost of the Techsat 21 system will be affected if a certain new technology is developed and validated in time to be incorporated into the TechSat 21 design, and whether or not the potential reward of waiting for the new technology is worth the potential schedule risk to the program.

The TechSat 21 trade space will be explored in greater detail in steps two through six of MMDOSA. The TechSat 21 GINA model is now complete, and step two of the MMDOSA methodology may begin.

### **5.3 MMDOSA Step 2 – Perform Univariate Studies**

Univariate studies may be used to find the general shape of a "design family" within the trade space. A design family is created by establishing a baseline design and then varying each parameter within the design vector across its entire range while holding all other components of the design vector at their baseline values. Table 5.9 lists the baseline design vector for the design family illustrated in Figure 5.31.

Figure 5.31 illustrates a sample TechSat 21 design family, with probability of detection (i.e. system performance) on the x-axis and lifecycle cost on the y-axis. Each dot represents a different TechSat 21 system architecture within the same design family. The black asterisk represents the baseline design (Table 5.9) for this particular family, and the diagonal lines are iso-CPF lines (i.e. lines of constant cost per unit probability of detection). Three separate types of relationships between the design vector parameters and system

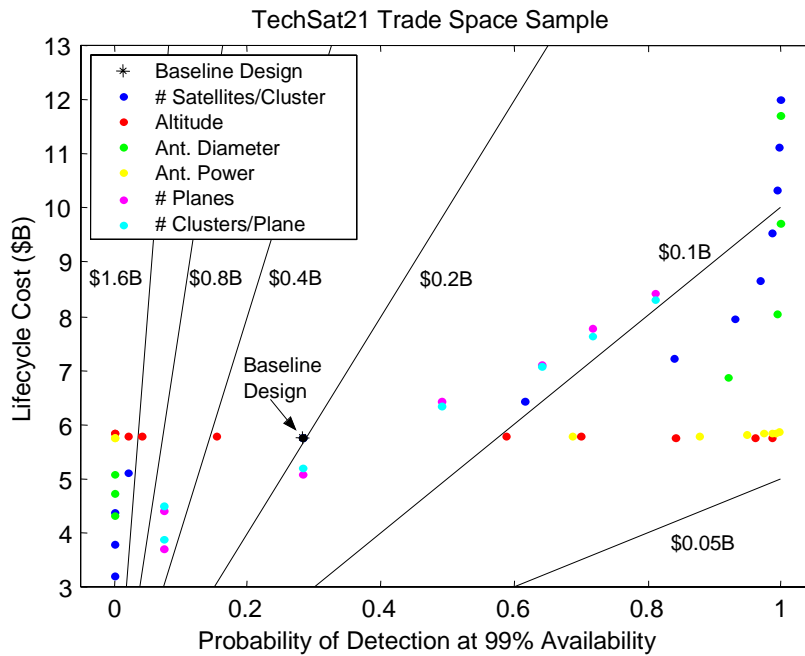


Figure 5.31 Example TechSat 21 Design Family

TABLE 5.9 Baseline TechSat 21 Design Vector

Design Vector	Value
# Satellites Per Cluster	8
Aperture Diameter	2 m
Radar Payload Power	300 W
Constellation Altitude	1000 km
# Clusters Per Orbital Plane	6
# Orbital Planes	6

metrics may be identified within this plot. The first relationship is represented by the blue dots, which show how performance and lifecycle cost vary as the number of satellites per cluster increases from four to sixteen. Starting with the left-most blue dot, increasing the number of satellites per cluster initially provides a dramatic increase in lifecycle cost, but with only a small corresponding increase in system performance. Beyond seven satellites per cluster, lifecycle cost continues to increase at the same rate, but now there is also a dramatic improvement in system performance. Beyond eleven satellites per cluster, lifecycle cost again continues to increase at the same rate, but now with only a marginal improve-

ment in system performance as it approaches its theoretical maximum of 100%. This results in the s-curve seen in the plot. Varying the aperture diameter (i.e. green dots) also produces a similar s-curve when plotted against system performance and lifecycle cost.

The second relationship is represented by the red dots, which show how performance and lifecycle cost vary as the altitude of the constellation increases from 500 km to 1500 km. Starting with the left-most red dot (i.e. 1500 km), system performance improves as altitude decreases, but apparently with no effect on lifecycle cost. In reality, lifecycle cost is decreasing with altitude as smaller, less costly launch vehicles are required to deploy and replenish the constellation. However, this cost savings is on the order of millions of dollars, and therefore does not show up well on the lifecycle cost scale in this plot, which is on the order of billions of dollars. A more pronounced negative slope would be seen if the range across which the altitude varied was greater than 1000 km. Varying the antenna transmission power (i.e. yellow dots) also produces a similar relationship in which system performance improves as transmission power increases, but with apparently no increase in lifecycle cost. In reality, lifecycle cost is increasing, but not with a magnitude observable in this plot due to the constant vector value of  $0.25 \text{ kg/m}^2$  for the areal density of the lithium ion solar arrays, which is the areal density the AFRL estimates will be available during the construction phase of this mission. Finally, both system performance and lifecycle cost improve linearly as the number of orbital planes (i.e. magenta dots) within the constellation and the number of clusters per orbital plane (i.e. cyan dots) increases. The plots in Figure 5.32 through Figure 5.41 illustrate several univariate studies executed on the TechSat 21 GINA model to verify model fidelity and gain a better feel for the trade space local to The Aerospace Corporation CDC baseline design (Table 5.7).

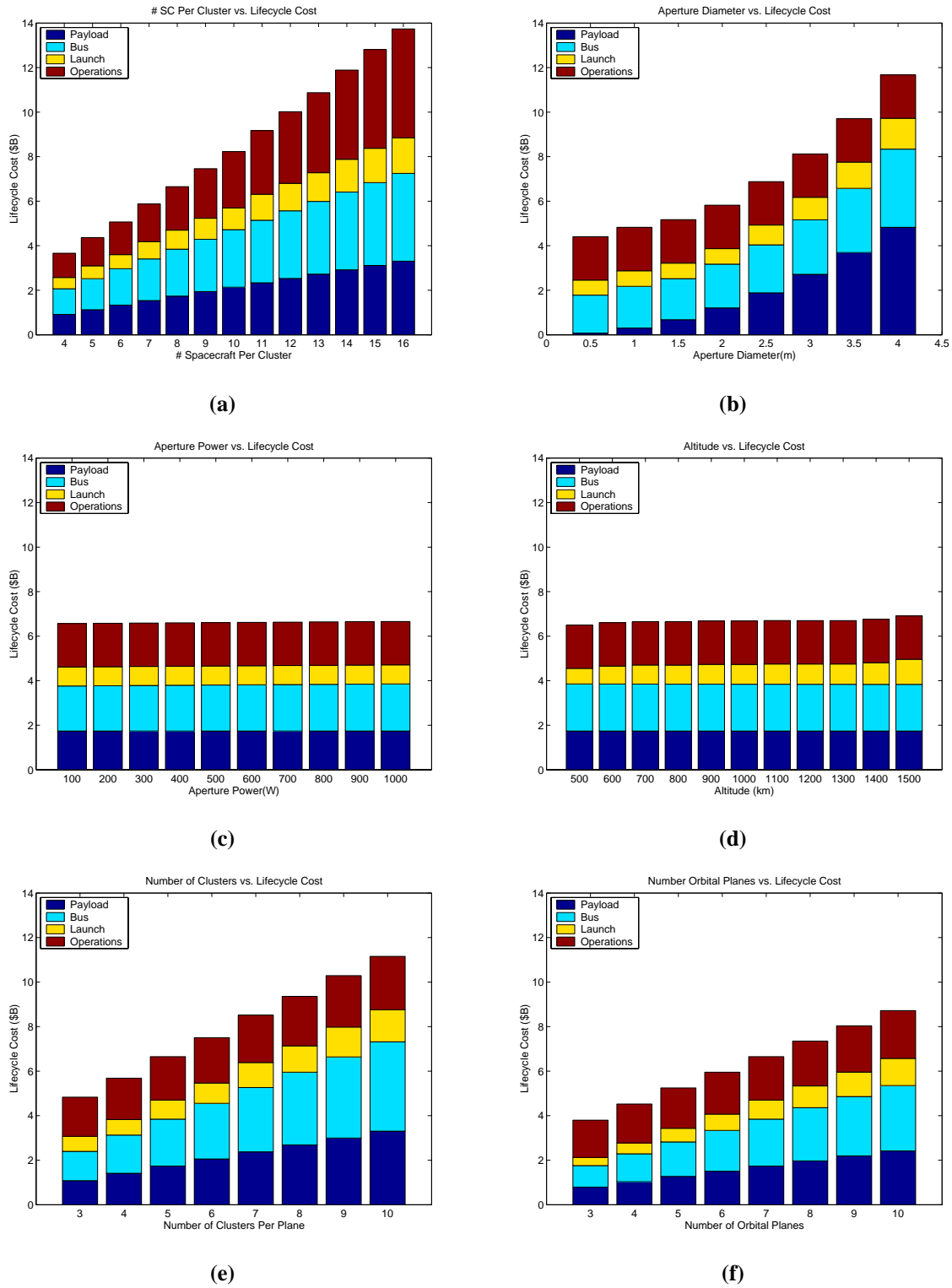
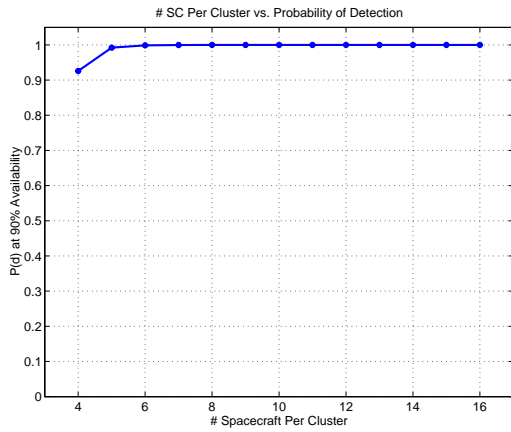
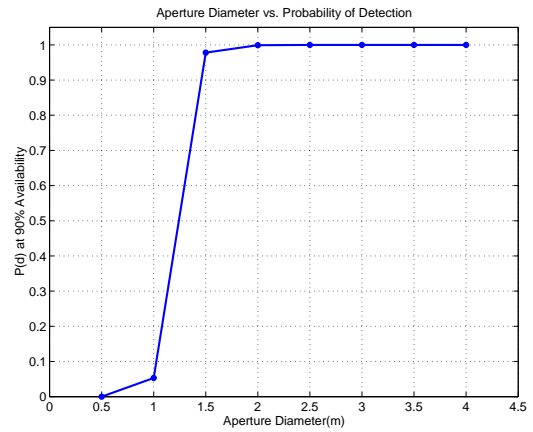


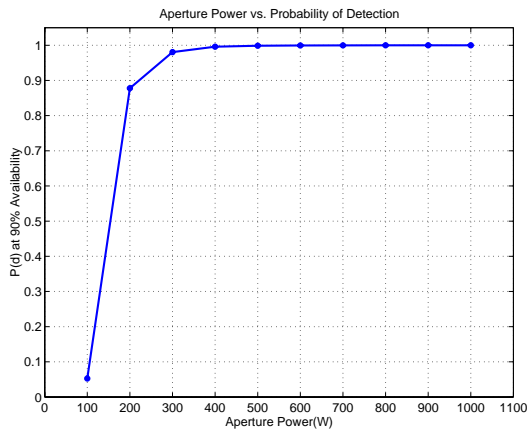
Figure 5.32 Lifecycle Cost TechSat 21 Univariate Studies



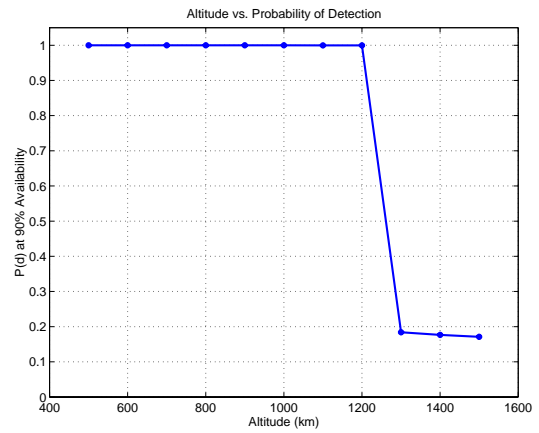
(a)



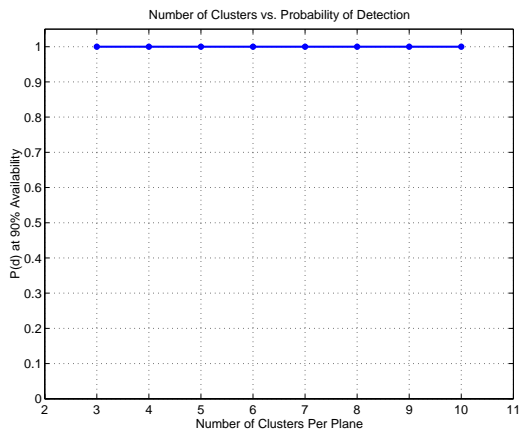
(b)



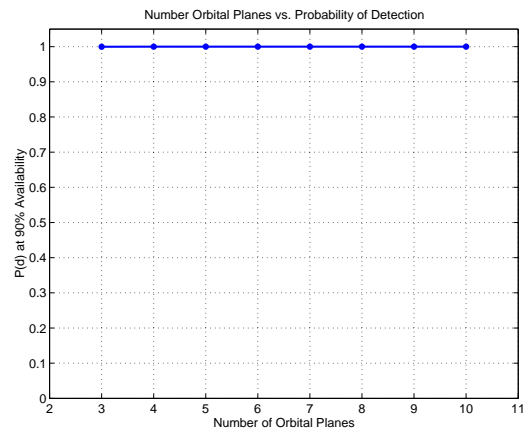
(c)



(d)

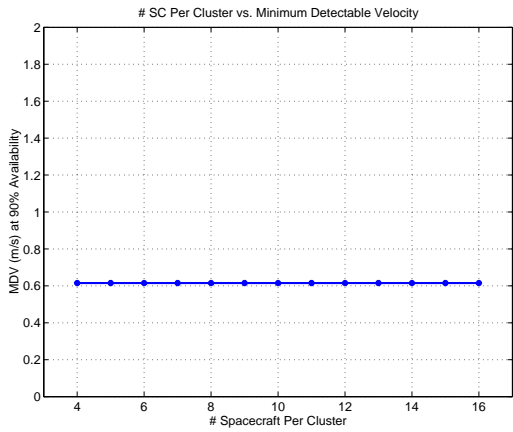


(e)

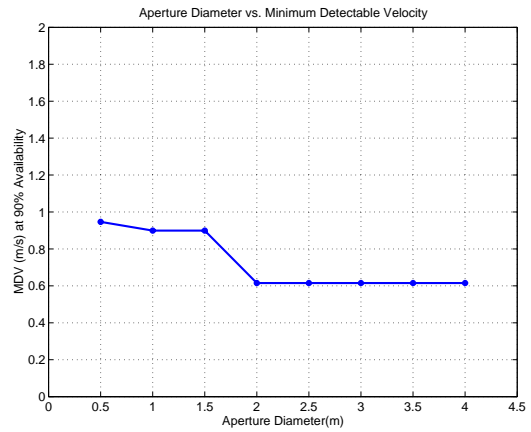


(f)

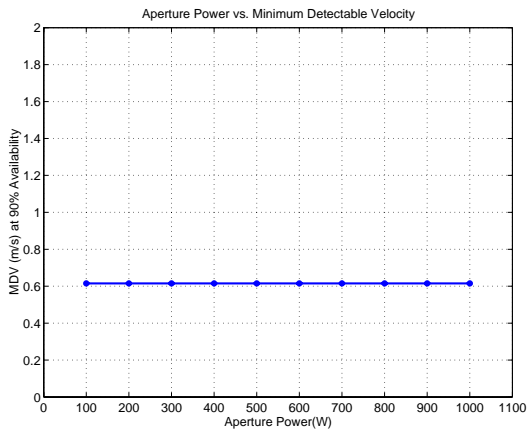
Figure 5.33 Probability of Detection TechSat 21 Univariate Studies



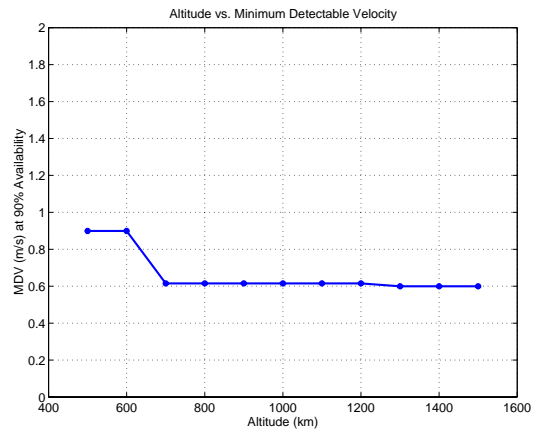
(a)



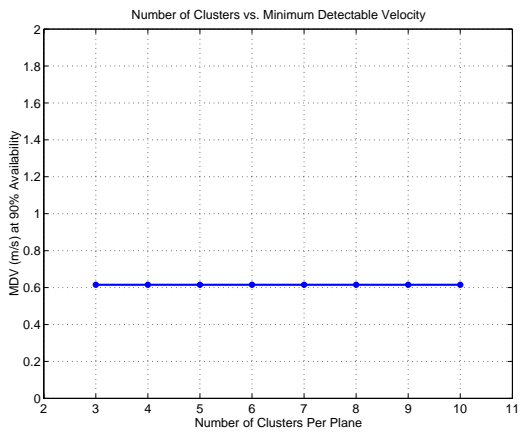
(b)



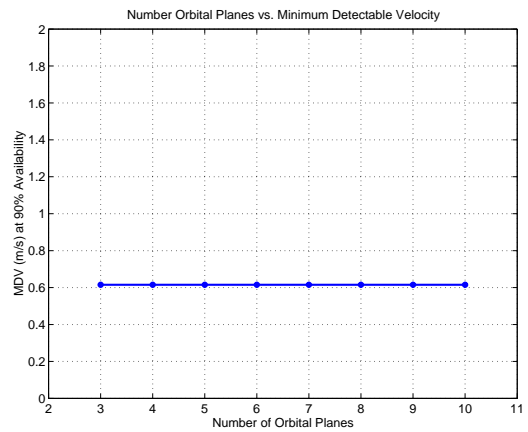
(c)



(d)



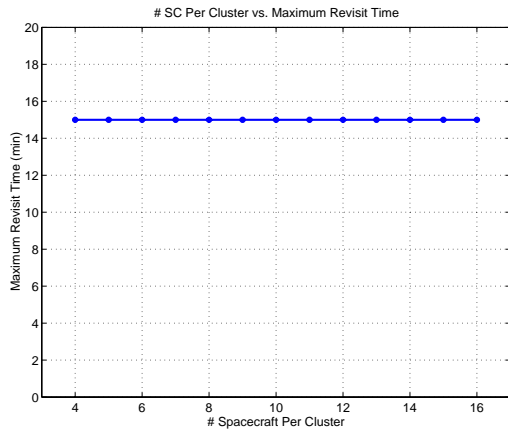
(e)



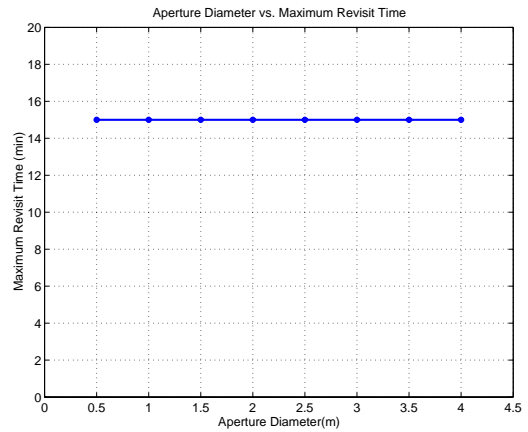
(f)

Figure 5.34 Minimum Detectable Velocity TechSat 21 Univariate Studies

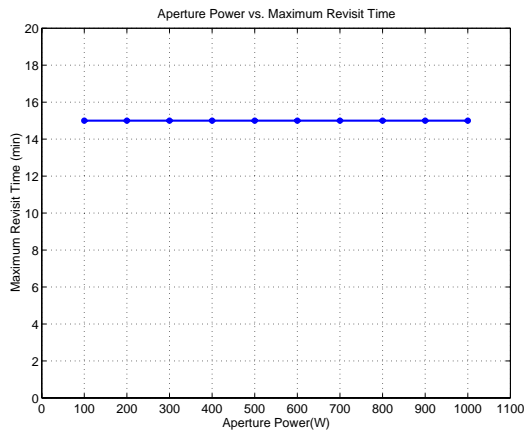




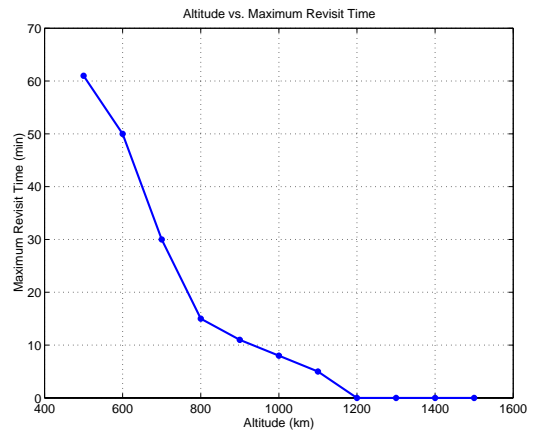
(a)



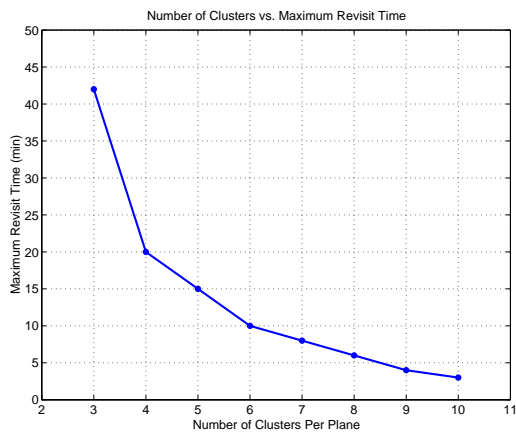
(b)



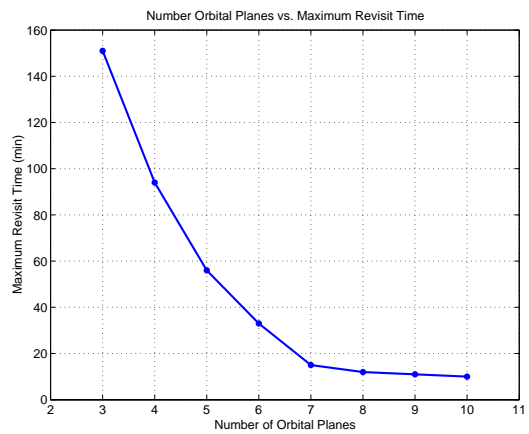
(c)



(d)

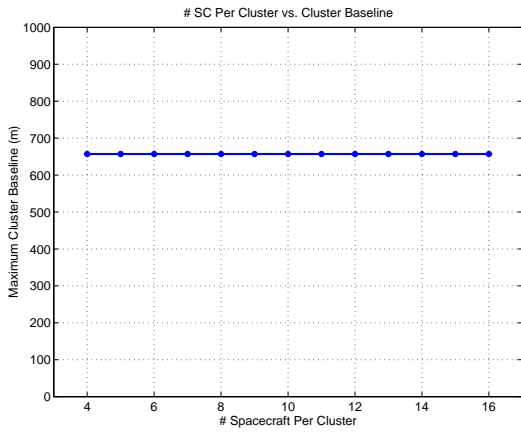


(e)

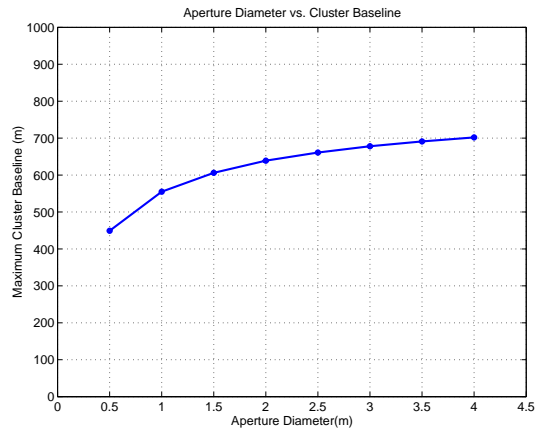


(f)

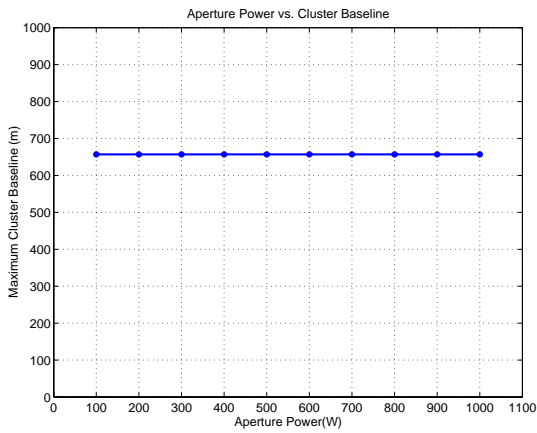
**Figure 5.35** Maximum Revisit Time TechSat 21 Univariate Studies



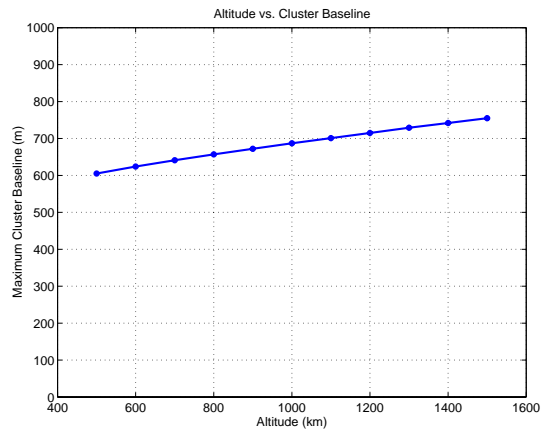
(a)



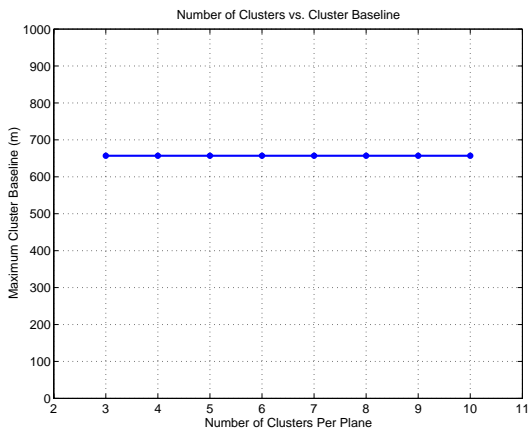
(b)



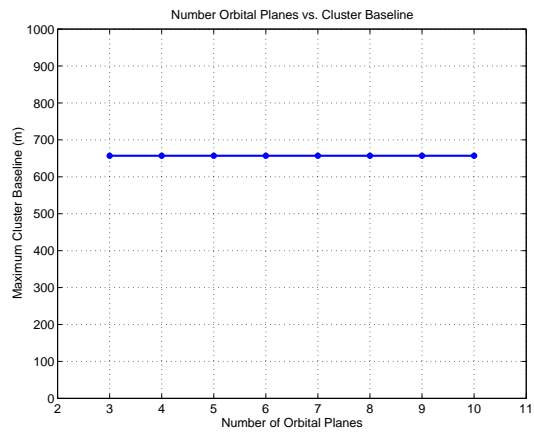
(c)



(d)

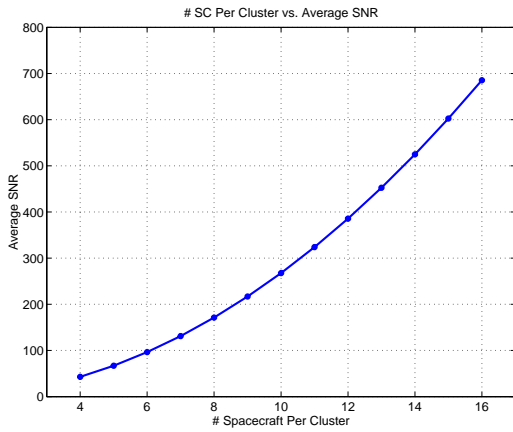


(e)

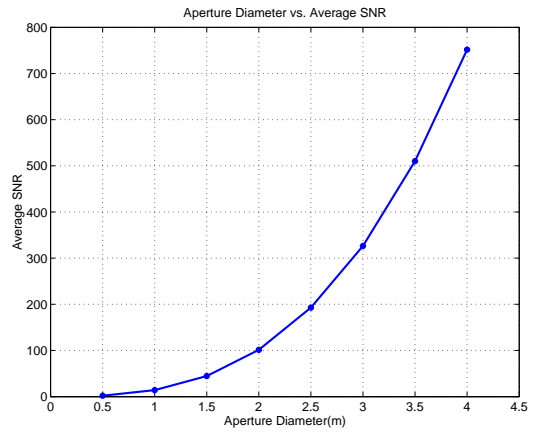


(f)

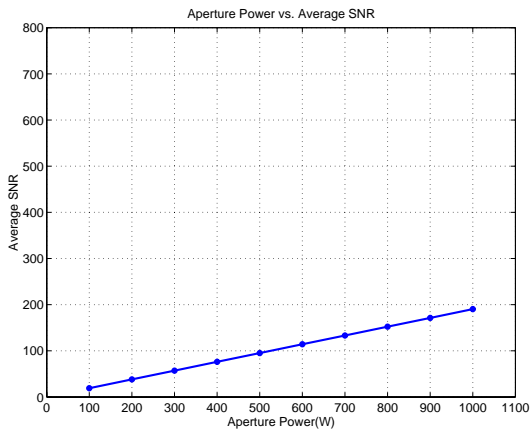
Figure 5.36 Maximum Cluster Baseline TechSat 21 Univariate Studies



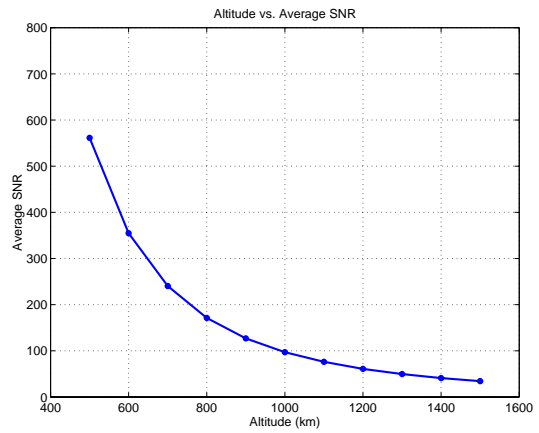
(a)



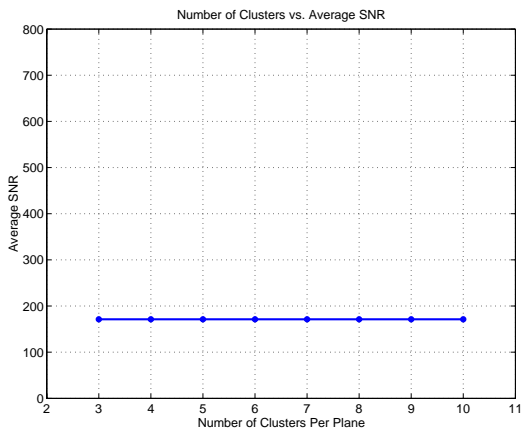
(b)



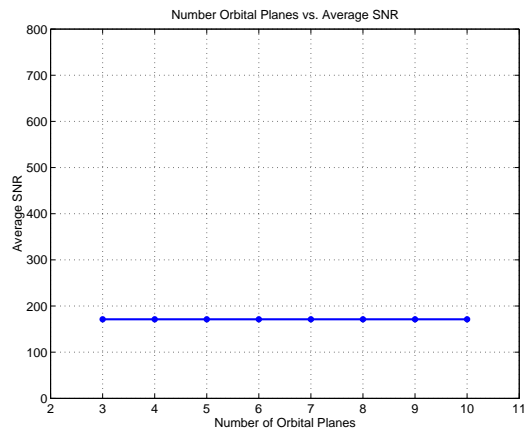
(c)



(d)

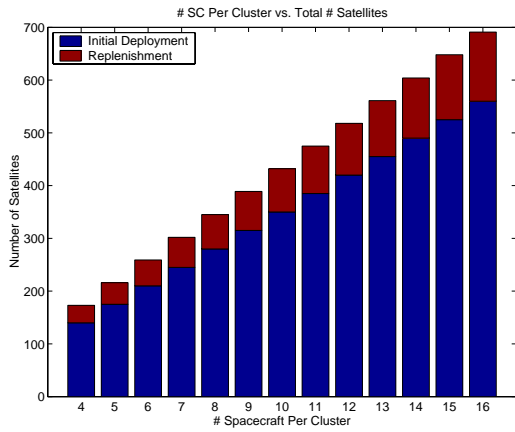


(e)

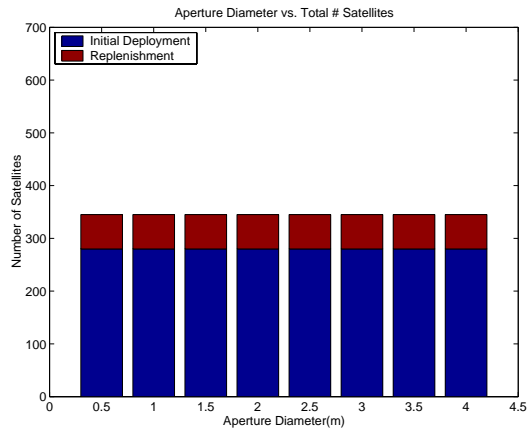


(f)

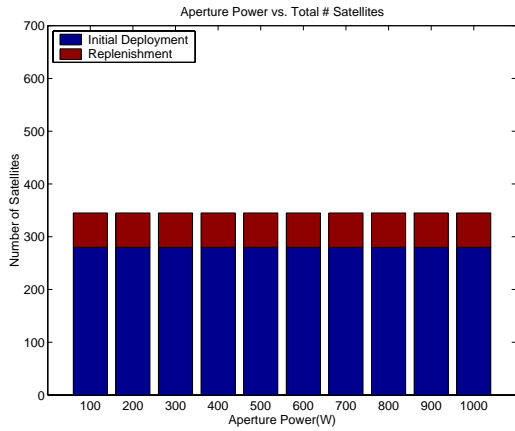
Figure 5.37 Average SNR TechSat 21 Univariate Studies



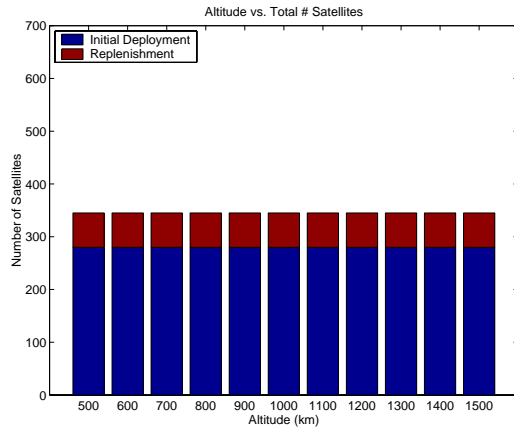
(a)



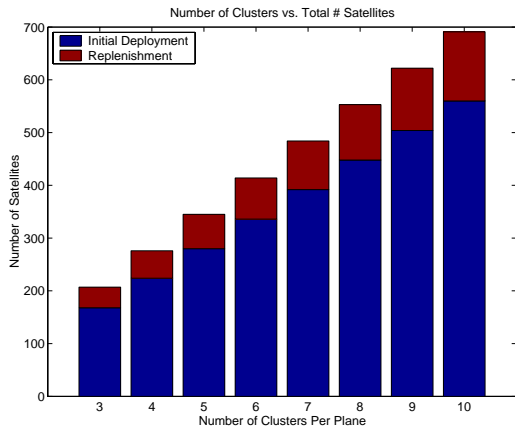
(b)



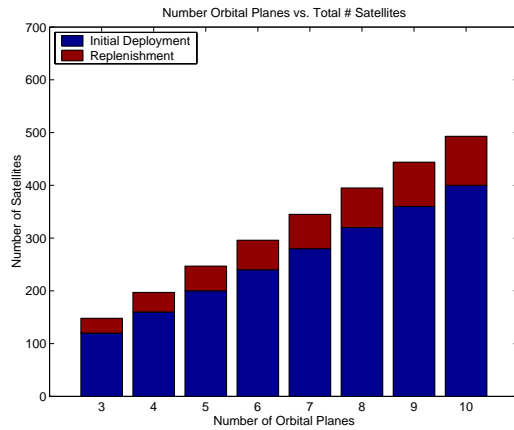
(c)



(d)

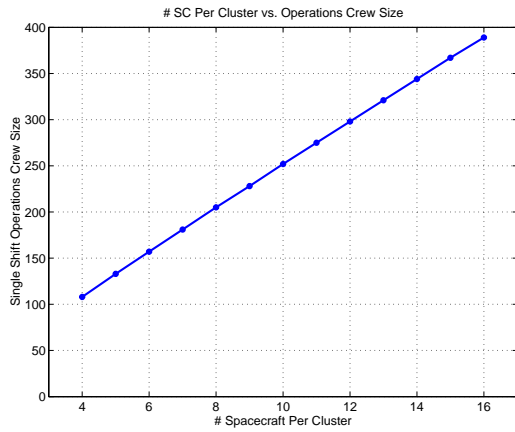


(e)

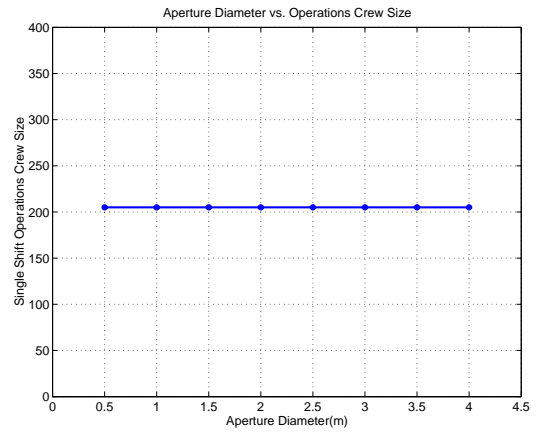


(f)

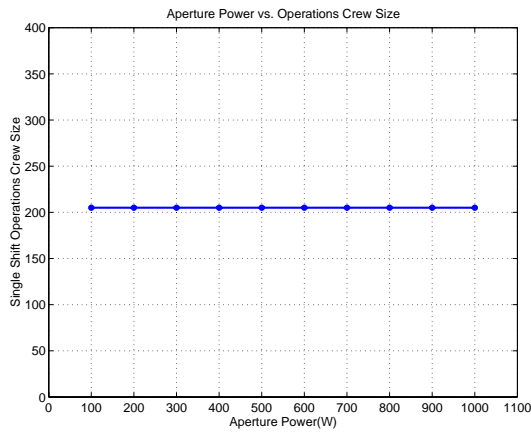
Figure 5.38 Number of Satellites TechSat 21 Univariate Studies



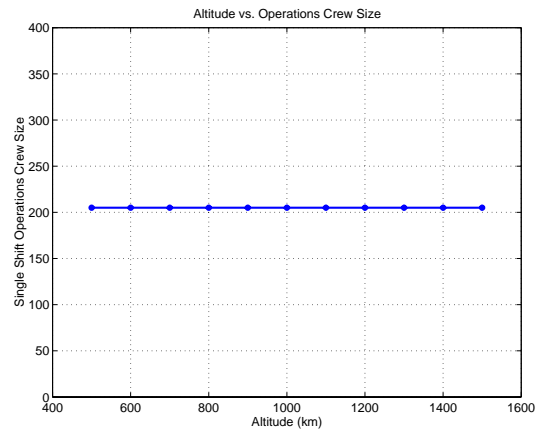
(a)



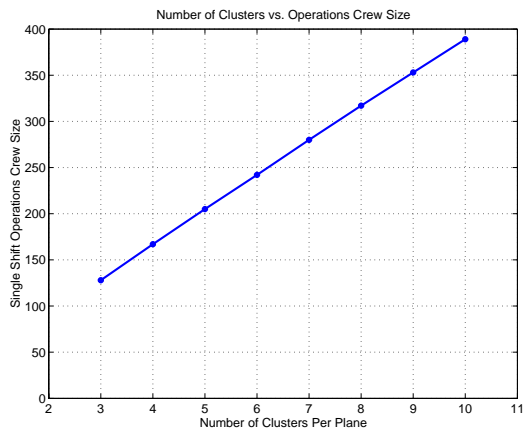
(b)



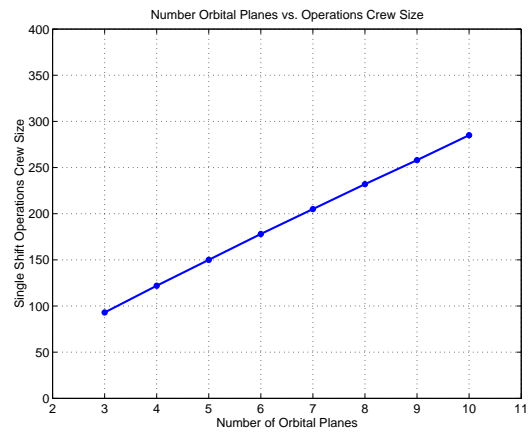
(c)



(d)



(e)



(f)

Figure 5.39 Single Shift Operations Crew Size TechSat 21 Univariate Studies

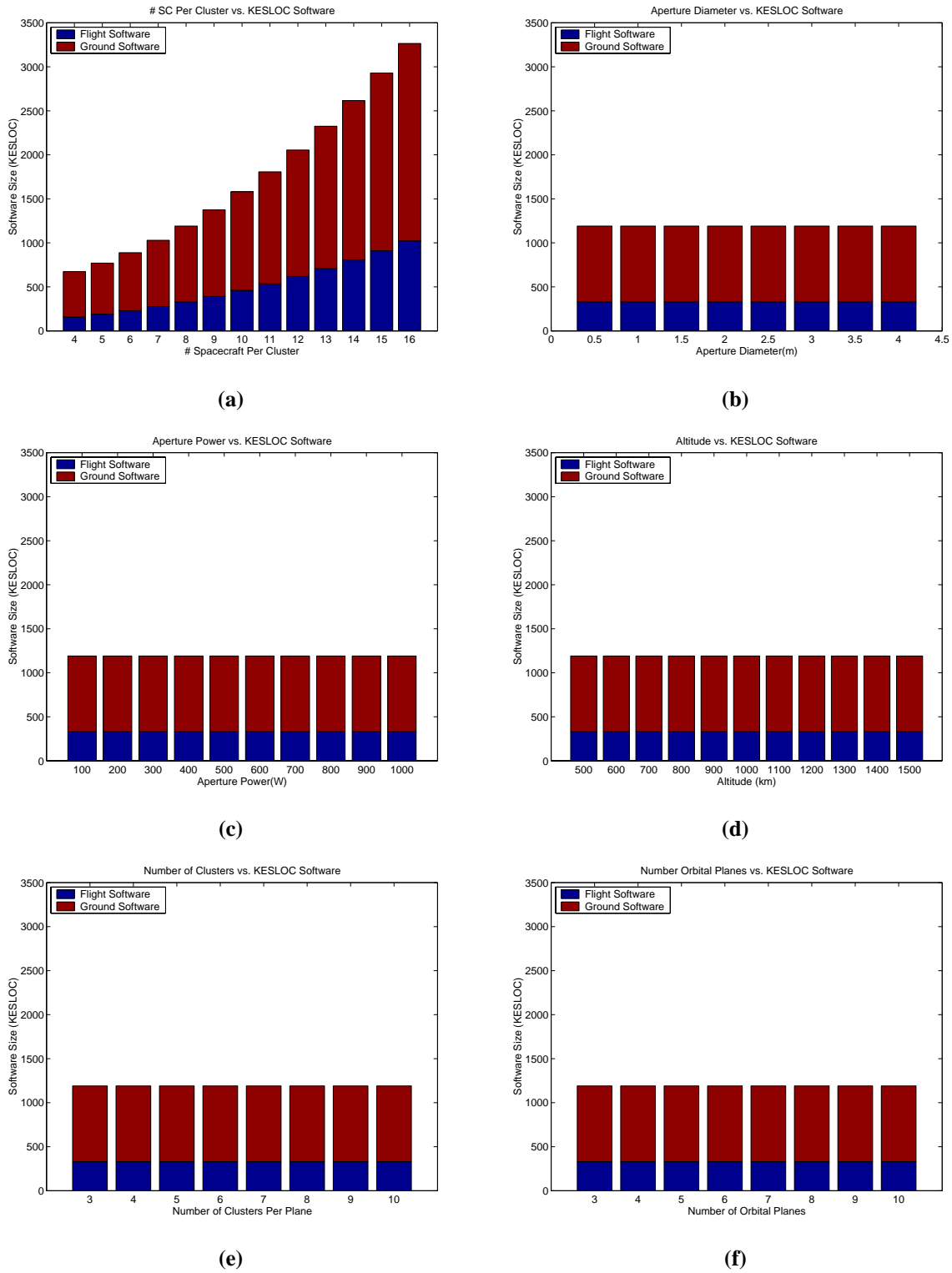
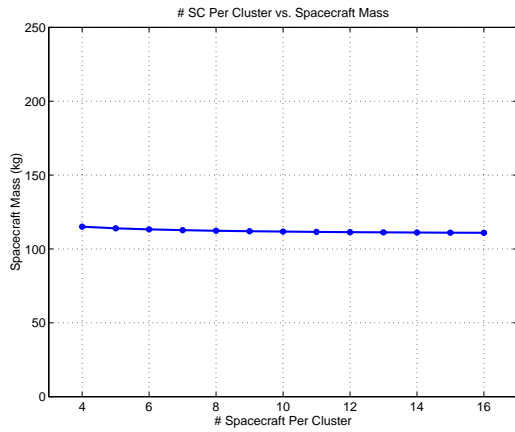
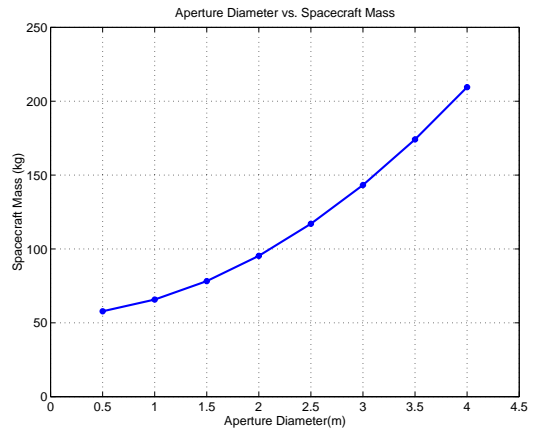


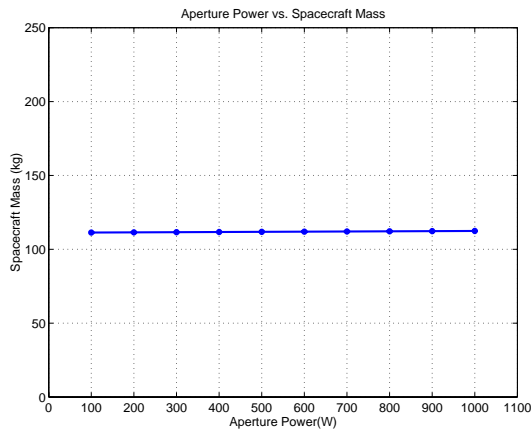
Figure 5.40 Software TechSat 21 Univariate Studies



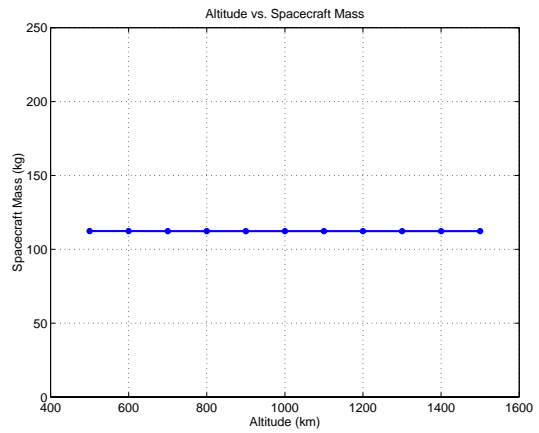
(a)



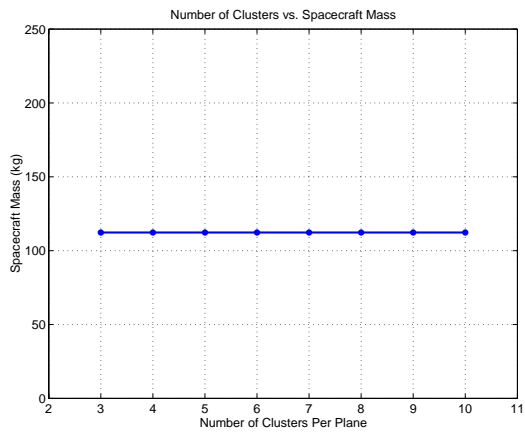
(b)



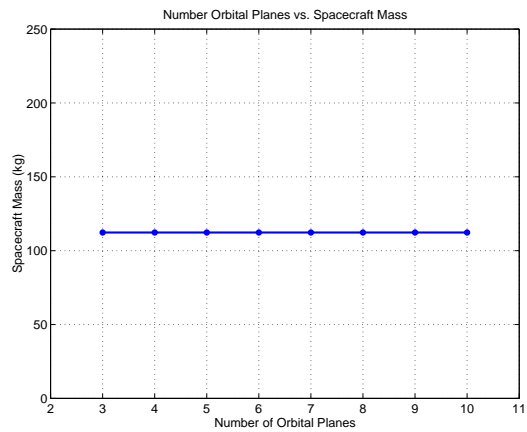
(c)



(d)



(e)



(f)

Figure 5.41 Spacecraft Mass TechSat 21 Univariate Studies

## 5.4 MMDOSA Step 3 – Take a Random Sample of the Trade Space

The importance of the  $\Delta$  parameter within the simulated annealing algorithm is discussed in Section 3.5. Ideally, this parameter would be computed with knowledge of the complete distribution of each metric which the DSS is being optimized with respect to. When exploring a large trade space, however, the  $\Delta$  parameter must be computed from the information gained from taking a random sample of the global trade space. For example, it currently takes approximately 12 hours to evaluate 2000 TechSat 21 system architectures on a 600 MHz Pentium II processor. At this computational speed, it would take 4393 hours (i.e. 183 days) to evaluate all 732,160 possible TechSat 21 system architectures!

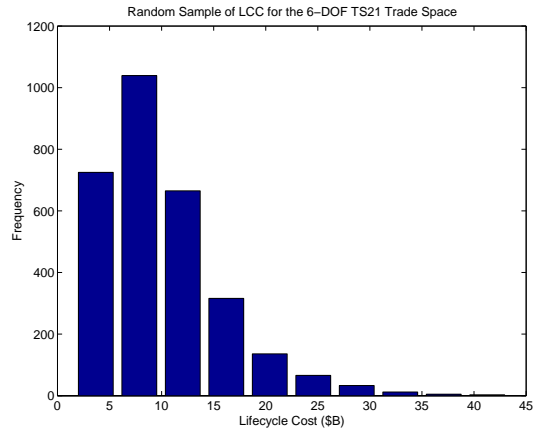
Within the TechSat 21 global trade space of 732,160 architectures, 3000 architectures – 0.41% of the trade space – were randomly sampled. Figure 5.42 illustrates the results from the random sample for each metric which the TechSat 21 design is simultaneously optimized with respect to in Section 5.5. Table 5.10 summarizes the statistics from this random sample. Please see Section 5.5 for details on how these statistics from the random sample are used to tailor the MDO search algorithms.

These statistics can be used to compute the 95% confidence interval (95%CI) for each parameter  $i$  in the objective function:

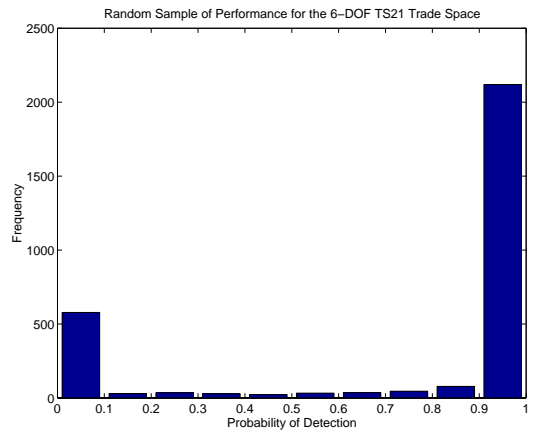
$$95\% \text{ CI} = \bar{x}_i \pm \frac{2v(x_i)}{\sqrt{n}} \quad (5.49)$$

where  $\bar{x}_i$  is the sample mean value for parameter  $i$ ,  $v(x_i)$  is the sample standard deviation for parameter  $i$ , and  $n$  is the size of the random sample. Table 5.11 lists the 95%CI of the mean value for the different architecture parameters that were measured in the random sample. This information tells the systems engineer what the likely characteristics of an "average" design will be. For TechSat 21, it can be said with a 95% confidence that a randomly selected set of TechSat 21 system architectures will exhibit on average a lifecycle

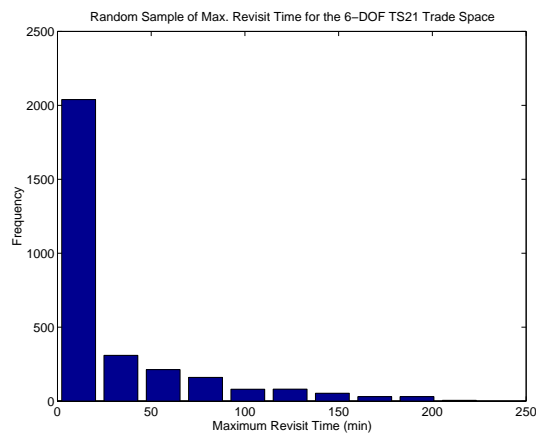




(a)



(b)



(c)

**Figure 5.42** Random Sample Results for Lifecycle Cost (a), Target Probability of Detection (b), and Maximum Revisit Time (c)

**TABLE 5.10** Random Sample Statistics of the TechSat 21 Global Trade Space

Parameter	Lifecycle Cost (\$B)	Target Probability of Detection	Maximum Revisit Time	Cost Per Unit Probability of Detection (\$M)
Maximum	43.4	1	224	$\infty$
Minimum	1.60	0	0	23.7
Range	41.8	1	224	$\infty$
Mean	10.0	0.762	26	$\infty$
Median	8.80	0.999	7	122
Standard Deviation	5.64	0.399	40	NA

cost of approximately \$10.0B, a target probability of detection of 76%, and a theater maximum revisit time of 26 minutes.

**TABLE 5.11** 95% CI's for the TechSat 21 Global Trade Space Mean

Parameter	95%CI
Lifecycle Cost	\$9.82 B - \$10.2 B
Target Probability of Detection	0.747 - 0.777
Maximum Revisit Time	24.6 min. - 27.6 min.

The systems engineer is not interested in creating an "average" design for the TechSat 21 system, however. Rather, the systems engineer wishes to find the best system architecture(s) based on the metric(s) of interest to the customer. Thus, the TechSat 21 architectures found via the MDO algorithms in step four of the MMDOSA methodology should rate much better than the 95%CI parameter values listed in Table 5.11. In fact, the data gathered in the random sample of the trade space (Table 5.10) can be used to place initial bounds on each of the design parameters to be optimized. For example, the minimum CPF TechSat 21 architectures found via MDO algorithms should yield a lower CPF than the minimum CPF value of \$23.7M in Table 5.10, which was the lowest CPF value measured in the random sample.

## 5.5 MMDOSA Step 4 – Formulate and Apply MDO Algorithm(s)

Four separate MDO approaches have been applied to the exploration of the TechSat 21 system trade space. These approaches are summarized in the test matrix in Table 5.12. In the first test case, a single objective simulated annealing algorithm is used to search the subset of the TechSat 21 trade space surrounding The Aerospace Corporation CDC baseline design (Table 5.7) that minimizes the cost per function metric. For the TechSat 21 GMTI mission, the CPF metric is defined as the cost per unit probability of detection.

**TABLE 5.12** MDO Test Matrix for the TechSat 21 Case Study

Test Case	Structure
<b>Approach 1</b>	
Type	Single Objective Optimization
Objective Function	Minimize CPF (i.e. Cost Per Unit Probability of Detection)
Trade Space Description	Aerospace Corporation Baseline Design (i.e. Fixed Constellation)
Trade Space Size	1040 system architectures
<b>Approach 2</b>	
Type	Single Objective Optimization
Objective Function	Minimize CPF (i.e. Cost Per Unit Probability of Detection)
Trade Space Description	TechSat 21 Global Trade Space
Trade Space Size	732,160 system architectures
<b>Approach 3</b>	
Type	Multiobjective Optimization
Objective Function	Minimize Lifecycle Cost & Maximize Probability of Detection
Trade Space Description	Aerospace Corporation Baseline Design (i.e. Fixed Constellation)
Trade Space Size	1040 system architectures
<b>Approach 4</b>	
Type	Multiobjective Optimization
Objective Function	Minimize Lifecycle Cost & Maximize Probability of Detection & Minimize Theater Revisit Time
Trade Space Description	TechSat 21 Global Trade Space
Trade Space Size	732,160 system architectures

In the second test case, the same single objective simulated annealing algorithm is applied to search the complete TechSat 21 global trade space (i.e. 732,160 separate system architectures). Complete enumeration of the trade space is not possible in this case, and thus there exists no way of verifying how close the family(s) of design solutions converged upon by the MDO algorithms after evaluating a fraction of this trade space are to the true optimal family(s) of solutions. However, by widening the scope and considering the global trade space, it is expected that design architectures with a lower CPF for the GMTI mission than those found in the first test case will be discovered.

In the third test case, the design scope is expanded from that of a single objective optimization problem to the more difficult multiobjective optimization problem in which the systems engineer simultaneously attempts to minimize lifecycle cost and maximize target probability of detection. At first, the scope is limited to the subset of the TechSat 21 trade space surrounding The Aerospace Corporation CDC baseline. In the fourth test case, the Pareto optimal set in the global TechSat 21 GMTI trade space of 732,160 separate system architectures is searched for. In this test case, a third dimension is added to the multiobjective optimization objective function – minimization of the theater revisit time.

### **5.5.1 Approach 1 – Single Objective Optimization of The Aerospace Corporation CDC Baseline Design Local Trade Space**

#### **Objective**

Figure 5.30 illustrates the cost and performance of all 1040 TechSat21 system architectures within the local trade space of The Aerospace Corporation CDC baseline design (Table 5.7). Complete enumeration of this local trade space yields a group of system architectures with a CPF of less than \$50M. The most cost-effective architecture within this family costs \$36.9M per unit probability of detection.

In this first stage of the test matrix, the objective is to find these most cost-effective TechSat 21 system architectures without completely enumerating the local trade space. This is done by applying the single objective simulated annealing algorithm described in

Section 3.5.1. The sole objective is to minimize the CPF metric, and the algorithm is terminated after 100 iterations (i.e. after evaluating 9.6% of the local trade space). Thus, this first design problem may be characterized as a single objective DSS architecture conceptual design problem.

**Problem Formulation**

This first TechSat 21 architecture design optimization problem may be formulated mathematically as:

$$\begin{array}{ll}
 \text{Objective:} & \text{Min} \left( \frac{\sum_{y=1}^{10} \Phi_y(\Gamma)}{\frac{\sum_{y=1}^{10} \Psi_{Pd_y}(\Gamma)}{10} \cdot (100)} \right) \\
 \\
 \text{Constraints:} & \text{Subject to} \\
 \text{Isolation} & \sigma_T \leq 10 \text{ dbsm} \\
 & MDV \leq 1.7 \text{ m/s} \\
 \text{Integrity} & \Psi_{Pd_y} \geq 0.95 \text{ for all } y \\
 \text{Rate} & T_{tr} \leq 15 \text{ min} \\
 & T_{tu} \leq 1 \text{ min} \\
 \text{Availability} & A_v \geq 0.95 \\
 \text{Bounds} & 4 \leq \gamma_1 \leq 16 \\
 & 0.5 \text{ m} \leq \gamma_2 \leq 4.0 \text{ m} \\
 & 100 \text{ W} \leq \gamma_3 \leq 1000 \text{ W} \\
 & \gamma_4 = 800 \text{ km} \\
 & \gamma_5 = 5 \\
 & \gamma_6 = 7
 \end{array} \tag{5.50}$$

where  $y$  is the year of the mission,  $\Phi$  is the system cost,  $\Psi_{Pd}$  is the target probability of detection,  $\sigma_T$  is the target radar cross section,  $MDV$  is the target minimum detection veloc-

ity,  $T_{tr}$  is the theater revisit time,  $T_{tu}$  is the target update time,  $A_v$  is the system availability, and  $\gamma$  are the design vector variables.

To evaluate a TechSat 21 architecture on the basis of the objective function and constraints in Eqn. 5.50, the design vector is entered into the TechSat 21 GINA model. The constellation design parameters are fixed within the local trade space of the Aerospace Corporation CDC design. Table 5.13 lists which design vector variables are free and fixed, along with the allowable range of values for each of the free variables.

**TABLE 5.13** Optimization Approach 1 Design Vector

Variable	Free/Fixed	Range/Value
# Satellites Per Cluster	Free	4 - 16
Aperture Diameter	Free	0.5 m - 4.0 m
Radar Payload Power	Free	100 W - 1000 W
Constellation Altitude	Fixed	800 km
# Clusters Per Orbital Plane	Fixed	5
# Orbital Planes	Fixed	7
<b>Total # Permutations</b>		<b>1040</b>

### Algorithm Design

A 2-DOF single objective simulated annealing algorithm was applied to the TechSat 21 CDC trade space optimization problem. Table 5.14 lists the cooling schedule used by each of the three variants of the core single objective simulated annealing algorithm. The initial system temperature was set to  $1 \times 10^8$  and was reduced by a factor of 0.97 after each iteration. The algorithm was terminated after the completion of 100 iterations.

**TABLE 5.14** Simulated Annealing Cooling Schedule

Cooling Schedule Parameters	Value
Initial System Temperature	$1 \times 10^8$ \$/Pd
Temperature Reduction Factor	0.97
Number Iterations	100

## Results

Figure 5.43 illustrates the local trade space of The Aerospace Corporation CDC TechSat 21 baseline design as determined from complete enumeration of all possible design vectors from Table 5.13. Table 5.15 lists the parameters for the design architectures found by the single objective simulated annealing algorithm that minimize the CPF metric. The family of most cost-effective architectures in the CDC trade space possesses four satellites per cluster, with each satellite exhibiting a 2.0 m - 2.5 m aperture diameter and a payload power of 700 W - 1000 W.

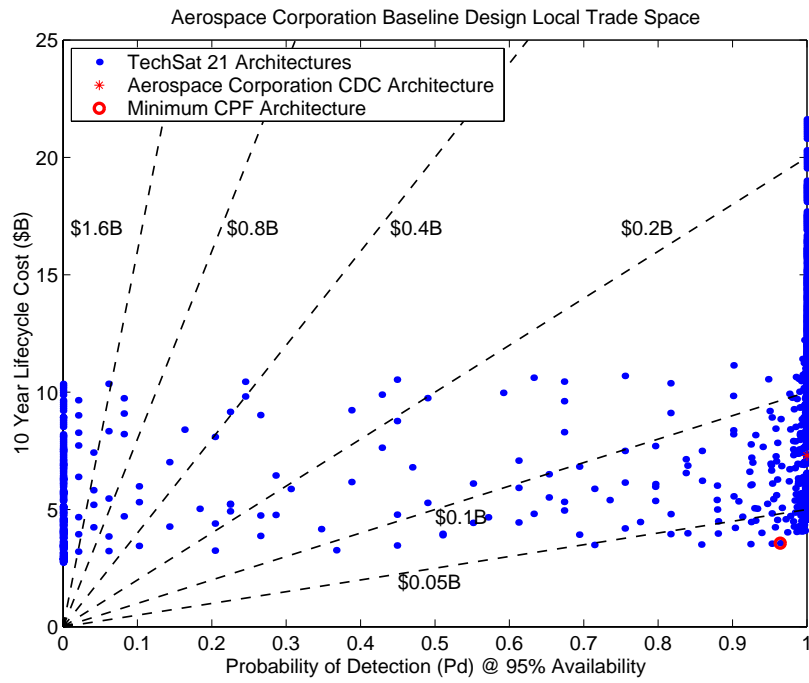
## Summary

In this section, the TechSat 21 conceptual design problem has been modeled as a single objective optimization problem, and a simulated annealing algorithm has been applied to search The Aerospace Corporation CDC baseline design local trade space for the family of most cost-effective architectures without completely enumerating the trade space. A family of architectures with a CPF nearly half that of the CDC baseline design were found. Now that this framework has been demonstrated within a local region of the TechSat 21 trade space, the next step is to apply it to the exploration of the global trade space.

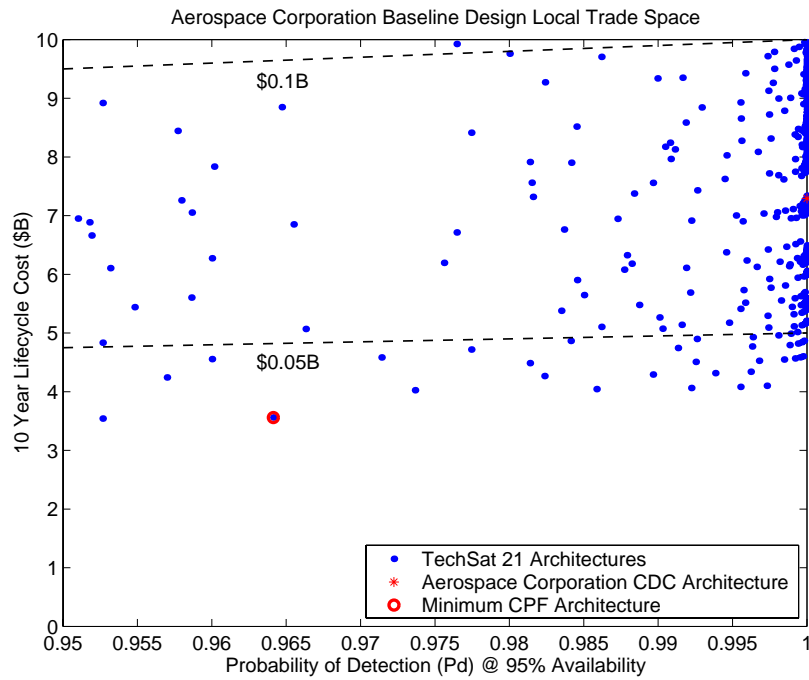
### 5.5.2 Approach 2 – Single Objective Optimization of the TechSat 21 Global Trade Space

#### Objective

In the second stage of the test matrix, the scope in the search for the most cost-effective architectures is expanded from The Aerospace Corporation CDC TechSat 21 baseline design local trade space to the TechSat 21 global trade space. The TechSat 21 global trade space contains 732,160 system architectures. Because complete enumeration of this vast trade space is not practical, it will not be possible to compare the MDO results with the unknown true optimal (i.e. minimum CPF) family of system architectures. However, it is expected that the architectures found by the MDO algorithms will be more cost-effective than those found via the random sample and local trade space search. A single objective



(a)



(b)

**Figure 5.43** Local Trade Space (a) and Local Trade Space Zoom-In (b) of The Aerospace Corporation CDC TechSat 21 Baseline Design



**TABLE 5.15** Minimum CPF Architectures Within the Local Trade Space

<b>Parameter</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>CDC</b>
# Satellites Per Cluster	4	4	4	4	4	8
Aperture Diameter (m)	2.0	2.0	2.0	2.0	2.5	2.5
Payload Power (W)	1000	900	800	700	800	1000
Constellation Altitude (km)	800	800	800	800	800	800
# Clusters Per Orbital Plane	5	5	5	5	5	5
# Orbital Planes	7	7	7	7	7	7
Lifecycle Cost (\$B)	3.56	3.54	3.52	3.51	4.06	7.29
Probability of Detection	0.964	0.953	0.924	0.858	0.992	0.999
Maximum Revisit Time (min)	15	15	15	15	15	15
Cost Per Function (\$M)	36.9	37.2	38.1	40.8	40.9	72.9

simulated annealing algorithm that strives to minimize the CPF metric will be applied. Further, the algorithm will be terminated after 2000 iterations (i.e. only 0.27% of the global trade space). This second conceptual design problem may also be characterized as a single objective DSS architecture design problem.

The mathematical formulation for the single objective global trade space search problem is:

$$\begin{array}{ll}
 \text{Objective:} & \text{Min} \left( \frac{\sum_{y=1}^{10} \Phi_y(\Gamma)}{\frac{\sum_{y=1}^{10} \Psi_{Pd_y}(\Gamma)}{10} \cdot (100)} \right) \\
 \\
 \text{Constraints:} & \text{Subject to} \\
 \text{Isolation} & \sigma_T \leq 10 \text{ dbsm} \\
 & MDV \leq 1.7 \text{ m/s} \\
 \text{Integrity} & \Psi_{Pd_y} \geq 0.95 \text{ for all } y \\
 \text{Rate} & T_{tr} \leq 15 \text{ min} \\
 & T_{tu} \leq 1 \text{ min} \\
 \text{Availability} & A_v \geq 0.95 \\
 \text{Bounds} & 4 \leq \gamma_1 \leq 16 \\
 & 0.5 \text{ m} \leq \gamma_2 \leq 4.0 \text{ m} \\
 & 100 \text{ W} \leq \gamma_3 \leq 1000 \text{ W} \\
 & 500 \text{ km} \leq \gamma_4 \leq 1500 \text{ km} \\
 & 3 \leq \gamma_5 \leq 10 \\
 & 3 \leq \gamma_6 \leq 10
 \end{array} \tag{5.51}$$

The global trade space search problem (Eqn. 5.51) design vector contains more potential degrees of freedom than the local trade space search problem (Eqn. 5.50). In the local trade space search problem formulation, the constellation parameters were fixed (Table 5.13), leaving a maximum of three degrees of freedom within the design vector. In the global trade space search problem formulation, every parameter within the design vector is free to be varied, yielding a design vector with a maximum of six degrees of freedom. Table 5.16 lists the range of allowable values for each variable in the six degree of freedom design vector.

**TABLE 5.16** Optimization Approach 2 Design Vector

Variable	Free/Fixed	Range/Value
# Satellites Per Cluster	Free	4 - 16
Aperture Diameter	Free	0.5 m - 4.0 m
Radar Transmission Power	Free	100 W - 1000 W
Constellation Altitude	Free	500 km - 1500 km
# Clusters Per Orbital Plane	Free	3 - 10
# Orbital Planes	Free	3 - 10
<b>Total # Permutations</b>		<b>732,160</b>

### Algorithm Design

A 2-DOF single objective simulated annealing algorithm was applied to the TechSat 21 global trade space optimization problem. Table 5.17 lists the cooling schedule used by this algorithm. The initial system temperature was set to  $1 \times 10^8$  and reduced by a factor of 0.998 after each iteration. The algorithm was terminated after the completion of 2000 iterations.

**TABLE 5.17** Simulated Annealing Cooling Schedule

Cooling Schedule Parameters	Value
Initial System Temperature	$1 \times 10^8$ \$/Pd
Temperature Reduction Factor	0.998
Number Iterations	2000

### Results

Table 5.18 lists the lowest CPF TechSat 21 system architectures converged upon by the single objective simulated annealing algorithm. Notice that the MDO algorithm found more cost-effective system architectures for the TechSat 21 mission within the global trade space than the system architectures found by complete enumeration of the local trade space around The Aerospace Corporation CDC baseline design. A lower CPF architecture was found in the random sample (Table 5.10), but this architecture is not a feasible solution to the conceptual design problem in Eqn. 5.51 because it violates the 15 minute the-

ater revisit time requirement. These results illustrate the power of using the single objective simulated annealing algorithm to search the global trade space of system architectures during the conceptual design phase of a DSS program. In this case, even a single application of the algorithm yielded a more cost-effective architecture than any previously considered.

**TABLE 5.18** Best TechSat 21 Architectures Found within the Global Trade Space

CPF (\$M/Pd)	LCC (\$B)	P(d) (%)	Theater Revisit Time (min.)	# Sats. Per Cluster	Apert. Diam. (m)	Radar Trans. Power (W)	Const. Altitude (km)	# Clusters Per Plane	# Orbital Planes
<b>Single Objective Simulated Annealing Algorithm</b>									
28.1	2.79	0.994	13	4	4	1000	1500	4	3
28.2	2.60	0.924	14	5	3.0	700	1300	4	3
29.8	2.94	0.987	14	5	3.5	700	1300	4	3
30.6	2.68	0.875	8	4	3.0	900	1200	4	4
33.2	3.14	0.945	14	4	2.5	900	1000	4	6
<b>The Aerospace Corporation Baseline Design Local Trade Space (Minimum CPF)</b>									
36.9	3.56	0.964	15	4	2.0	1000	800	5	7
<b>Random Sample (Minimum)</b>									
23.7	2.37	99.9	169	6	1.5	900	500	3	4

## Summary

In this section, the single objective simulated annealing algorithm has been applied to search the TechSat 21 global trade space for the most cost-effective system architectures without complete enumeration of the enormous trade space. After only one trial, the MDO algorithm found more cost-effective system architectures than any architecture that had previously been identified by the AFRL or The Aerospace Corporation. These results illustrate the power of the single objective simulated annealing algorithm for DSS conceptual design.

### **5.5.3 Approach 3 – Multiobjective Optimization of The Aerospace Corporation CDC Baseline Design Local Trade Space**

#### **Objective**

In this third stage of the test matrix, the TechSat 21 conceptual design problem is modeled as a multiobjective optimization problem with the intention of finding the Pareto optimal set of system architectures in The Aerospace Corporation CDC baseline design local trade space without completely enumerating the trade space. This is done by applying the multiobjective multiple solution simulated annealing algorithm developed in Section 3.5.2, which attempts to satisfy simultaneously the competing objectives of minimizing lifecycle cost and maximizing target probability of detection. The algorithm is terminated after 100 iterations (i.e. 9.6% of the local trade space). This third conceptual design problem may be characterized as a multiobjective DSS architecture design problem.

#### **Problem Formulation**

This multiobjective conceptual design optimization problem may be formulated mathematically as:

$$\begin{array}{ll}
 \text{Objective:} & \text{Min} \left( \sum_{y=1}^{10} \Phi_y(\Gamma) \right) \text{ AND Max} \left( \frac{\sum_{y=1}^{10} \Psi_{Pd_y}(\Gamma)}{10} \cdot (100) \right) \\
 \\
 \text{Constraints:} & \text{Subject to} \\
 \text{Isolation} & \sigma_T \leq 10 \text{ dbsm} \\
 & MDV \leq 1.7 \text{ m/s} \\
 \text{Integrity} & \Psi_{Pd_y} \geq 0.95 \text{ for all } y \\
 \text{Rate} & T_{tr} \leq 15 \text{ min} \\
 & T_{tu} \leq 1 \text{ min} \\
 \text{Availability} & A_v \geq 0.95 \\
 \text{Bounds} & 4 \leq \gamma_1 \leq 16 \\
 & 0.5 \text{ m} \leq \gamma_2 \leq 4.0 \text{ m} \\
 & 100 \text{ W} \leq \gamma_3 \leq 1000 \text{ W} \\
 & \gamma_4 = 800 \text{ km} \\
 & \gamma_5 = 5 \\
 & \gamma_6 = 7
 \end{array} \tag{5.52}$$

To evaluate a TechSat 21 system architecture on the basis of the objective function and constraints in Eqn. 5.52, the design vector is entered into the TechSat 21 GINA model. The constellation design parameters are fixed within the local trade space of The Aerospace Corporation CDC design. Table 5.19 lists which design vector variables are free and fixed, along with the allowable range of values for each of the free variables.

### Algorithm Design

In addition to the change in the mathematical formulation, the multiobjective optimization problem also requires a change in the internal mechanics of the simulated annealing algorithm. As explained in Section 3.5,  $\Delta$  is a parameter that is used to determine the probability with which the algorithm may move to a less optimal solution with the intent of

**TABLE 5.19** Optimization Approach 3 Design Vector

Variable	Free/Fixed	Range/Value
# Satellites Per Cluster	Free	4 - 16
Aperture Diameter	Free	0.5 m - 4.0 m
Radar Transmission Power	Free	100 W - 1000 W
Constellation Altitude	Fixed	800 km
# Clusters Per Orbital Plane	Fixed	5
# Orbital Planes	Fixed	7
<b>Total # Permutations</b>		<b>1040</b>

escaping local optima within the nonconvex trade space. In the single objective case,  $\Delta$  is simply the difference in the objective function value between successive steps of the algorithm. For multiobjective problems, however,  $\Delta$  must take into account the difference in successive steps between all  $N$  dimensions of the objective function.

The formula for computing the  $\Delta$  parameter for a multiobjective optimization problem with  $N$  dimensions in the objective function (i.e.  $N$  design goals or  $N$  decision criteria) is:

$$\Delta = \frac{\sum_{n=1}^N \frac{E_n(\Gamma_{i+1}) - E_n(\Gamma_i)}{\delta_n}}{N} \quad (5.53)$$

where  $i$  is the iteration number within the simulated annealing algorithm and  $\delta_n$  is the maximum range in dimension  $n$  observed in the random sample. Depending upon the distribution of the random sample,  $\delta_n$  may be replaced by the  $1\sigma_n$  range,  $2\sigma_n$  range, or  $3\sigma_n$  range for each dimension  $n$  in the objective function.

For the multiobjective optimization of The Aerospace Corporation CDC baseline design local trade space, a random sample is not necessary because the local trade space has already been completely enumerated. The results of this complete enumeration for each dimension of the objective function in Eqn. 5.52 are summarized in Table 5.20.

**TABLE 5.20** Objective Function Statistics From Complete Enumeration of The Aerospace Corporation CDC Baseline Design Local Trade Space

Parameter	Lifecycle Cost (\$B)	Target Probability of Detection
Maximum	21.7	1.00
Minimum	2.74	0.00
Range	18.9	1.00
Mean	9.01	0.758
Median	8.35	0.999
Standard Deviation	4.11	0.405

Using the information in Table 5.20, the simulated annealing  $\Delta$  parameter for the multiobjective optimization problem in Eqn. 5.52 may now be computed as:

$$\Delta = \frac{\frac{\Phi(\Gamma_{i+1}) - \Phi(\Gamma_i)}{\$18.9 \times 10^9} + \frac{\Psi_{Pd}(\Gamma_i) - \Psi_{Pd}(\Gamma_{i+1})}{1}}{2} \quad (5.54)$$

A 2-DOF variant of the multiobjective multiple solution simulated annealing algorithm was used. Table 5.21 lists the cooling schedule of the core multiobjective simulated annealing algorithm. Initial system temperature was set to one and was reduced by a factor of 0.97 after each iteration. The algorithm was terminated after the completion of 100 iterations.

**TABLE 5.21** Simulated Annealing Cooling Schedule

Cooling Schedule Parameters	Value
Initial System Temperature	1
Temperature Reduction Factor	0.97
Number Iterations	100

## Results

Figure 5.44a illustrates the Pareto optimal front within The Aerospace Corporation CDC baseline design local trade space. The 2-DOF multiobjective, multiple solution simulated annealing algorithm successfully found these Pareto optimal architectures after evaluating



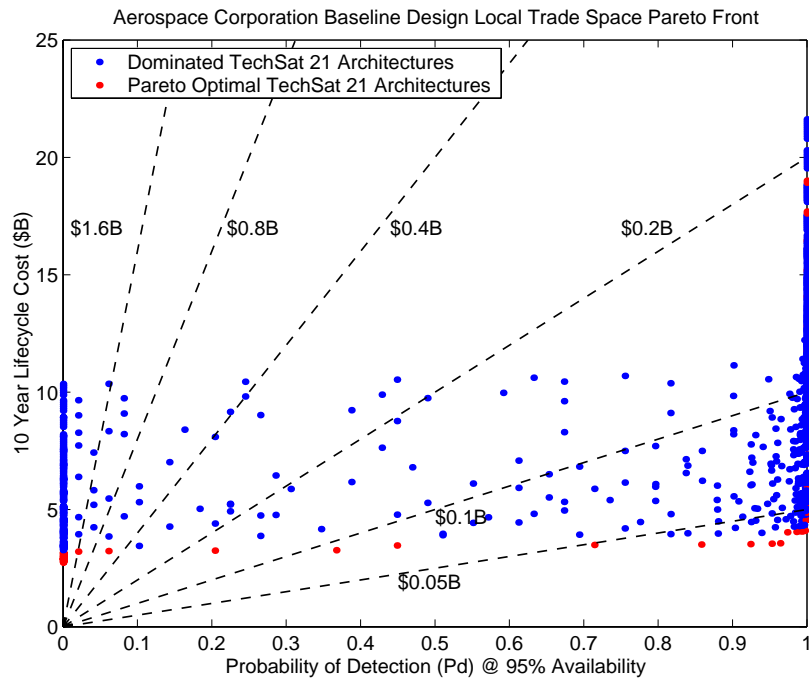
only a fraction of the CDC trade space. Within this Pareto optimal set, the systems engineer cannot increase the target probability of detection without increasing lifecycle cost. Table 5.22 lists the Pareto optimal architectures in Figure 5.44b.

### **Summary**

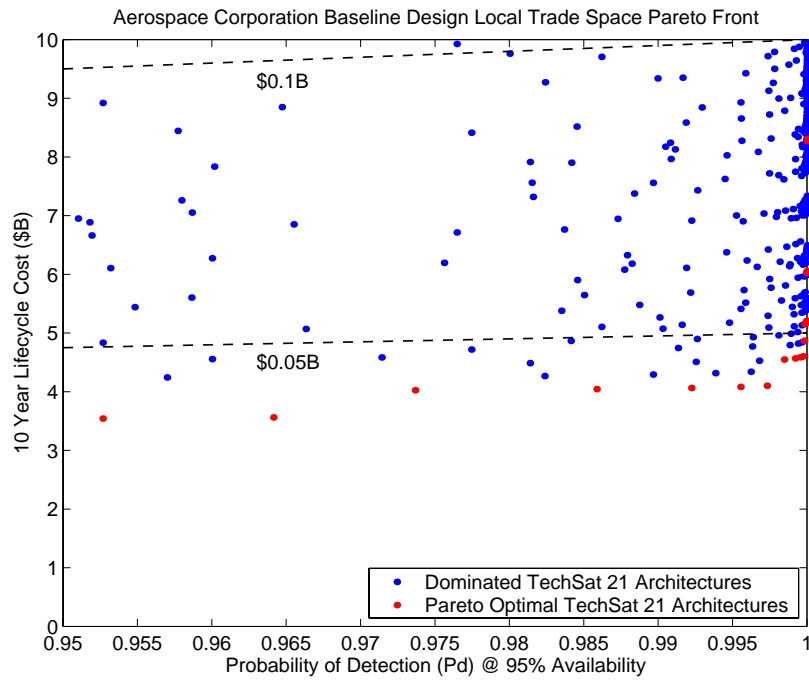
In this section, the TechSat21 architecture conceptual design problem has been modeled as a multiobjective optimization problem, and the multiobjective multiple solution simulated annealing algorithm has been applied to search The Aerospace Corporation CDC baseline design local trade space for the Pareto optimal set of system architectures without completely enumerating the trade space. Now that this multiobjective framework has been demonstrated within a local region of the TechSat 21 trade space, the next step is to apply it to the exploration of the global trade space.

### **5.5.4 Approach 4 – Multiobjective Optimization of the TechSat 21 Global Trade Space**

In the fourth stage of the test matrix, the scope in the search for the Pareto optimal set of architectures is expanded from The Aerospace Corporation CDC TechSat 21 baseline design local trade space to the TechSat 21 global trade space. The TechSat 21 global trade space contains 732,160 system architectures. Because complete enumeration of this vast trade space is not practical, it will not be possible to compute the true Pareto optimal set. Therefore it is not possible to compare the MDO results with the true Pareto boundary. The design problem becomes even more complicated with the addition of a third dimension to the objective function. In addition to attempting simultaneously to minimize lifecycle cost and maximize target probability of detection, the algorithm will also attempt to minimize the theater revisit time. This will be done by applying a multiobjective multiple solution simulated annealing algorithm which terminates after 2000 iterations (i.e. after evaluating only 0.27% of the global trade space). As was the case in the third stage of the test matrix, this fourth design problem may also be characterized as a multiobjective DSS system conceptual design problem.



(a)



(b)

**Figure 5.44** The Aerospace Corporation CDC Baseline Design Local Trade Space Pareto Front (a) and Pareto Front Zoom-In (b)

**TABLE 5.22** Pareto Optimal TechSat 21 Architectures from Figure 5.44b

Architecture	1	2	3	4	5	6	7	8	9
# Satellites Per Cluster	4	4	4	4	4	4	4	4	4
Aperture Diameter	2.0	2.0	2.5	2.5	2.5	2.5	2.5	3.0	3.0
Radar Transmission Power	900	1000	600	700	800	900	1000	700	800
Target Probability of Detection	0.953	0.964	0.974	0.986	0.992	0.996	0.997	0.998	0.999
Lifecycle Cost (\$B)	3.54	3.56	4.02	4.04	4.06	4.08	4.10	4.55	4.57
Max. Theater Revisit Time (min.)	15	15	15	15	15	15	15	15	15

**Problem Formulation**

The fourth TechSat 21 conceptual design optimization problem may be formulated mathematically as:

Objective: 
$$\text{Min} \left( \sum_{y=1}^{10} \Phi_y(\Gamma) \right) \text{ AND } \text{Max} \left( \frac{\sum_{y=1}^{10} \Psi_{Pd_y}(\Gamma)}{10} \cdot (100) \right) \text{ AND } \text{Min}(T_{tr})$$

Constraints: Subject to

Isolation  $\sigma_T \leq 10$  dbsm

$MDV \leq 1.7$  m/s

Integrity  $\Psi_{Pd_y} \geq 0.95$  for all  $y$  (5.55)

Rate  $T_{tu} \leq 1$  min

Availability  $A_v \geq 0.95$

Bounds  $4 \leq \gamma_1 \leq 16$

$0.5 \text{ m} \leq \gamma_2 \leq 4.0 \text{ m}$

$100 \text{ W} \leq \gamma_3 \leq 1000 \text{ W}$

$500 \text{ km} \leq \gamma_4 \leq 1500 \text{ km}$

$3 \leq \gamma_5 \leq 10$

$3 \leq \gamma_6 \leq 10$

Notice that this problem formulation is similar to the previous multiobjective optimization problem in Eqn. 5.52; with the exception that the theater revisit time  $T_{tr}$  has moved from being a rate requirement to being the third dimension of the objective function, and the design vector in Eqn. 5.55 contains more potential degrees of freedom. In the local trade space search problem formulation (Eqn. 5.52), the constellation parameters were fixed (Table 5.19), leaving a maximum of three degrees of freedom within the design vector. In the global trade space search problem formulation (Eqn. 5.55), every parameter within the design vector is free to be varied, yielding a design vector with a maximum of six degrees of freedom. Table 5.23 lists the range of allowable values for each variable in the six degree of freedom design vector.

**TABLE 5.23** Optimization Approach 4 Design Vector

Variable	Free/Fixed	Range/Value
# Satellites Per Cluster	Free	4 - 16
Aperture Diameter	Free	0.5 m - 4.0 m
Radar Transmission Power	Free	100 W - 1000 W
Constellation Altitude	Free	500 km - 1500 km
# Clusters Per Orbital Plane	Free	3 - 10
# Orbital Planes	Free	3 - 10
<b>Total # Permutations</b>		<b>732,160</b>

### Algorithm Design

For The Aerospace Corporation CDC baseline design local trade space multiobjective optimization problem (Eqn. 5.52), the  $\delta_n$  parameter in Eqn. 5.53 was computed for each dimension  $n$  of the objective function with the knowledge gained from complete enumeration of the local trade space. When exploring the global trade space, however, the  $\Delta$  parameter must be computed from the information gained from taking a random sample of the global trade space. Table 5.10 in Section 5.4 summarizes the objective function statistics from a random sample of 3000 TechSat 21 system architectures within the global trade space of 732,160 architectures.

In part three of the test matrix, the observed range for each parameter in the objective function was used for the  $\delta_n$  parameter in the computation of  $\Delta$  in Eqn. 5.54. A potential problem with this procedure is that the range may be determined by only a handful of obscure data points. This in turn skews the probability in Eqn. 3.30 that the simulated annealing algorithm will move temporarily to a less optimal solution as it strives to escape local optima.

To remedy this problem, the  $+2\sigma$  value will be used rather than the entire range for the  $\delta_n$  parameter in the computation of  $\Delta$ . The  $+2\sigma$  value for each dimension of the objective function implies that 97.5% of all possible system architectures exhibit a value less than  $+2\sigma$  along that particular dimension. The  $\delta_n$  parameter represents a range around which each objective function will be normalized, and may now be computed as:

$$\delta_n = \bar{x}_n + 2\sigma_n - x_{n_{\min}} \tag{5.56}$$

where  $\sigma_n$  is the sample standard deviation along dimension  $n$  and  $x_{n_{\min}}$  is the minimum value observed in the random sample along dimension  $n$ . Table 5.24 contains the results of Eqn. 5.56 for each dimension of the objective function in Eqn. 5.55.

**TABLE 5.24**  $\delta_n$  Values for the Multiobjective Optimization of the TechSat 21 Global Trade Space

Objective Function Dimension	$\delta_n$
Lifecycle Cost	\$19.7 B
Target Probability of Detection	1
Maximum Revisit Time	106 min.

With this information, the simulated annealing  $\Delta$  parameter may be specifically computed for the multiobjective optimization problem in Eqn. 5.55 as:

$$\Delta = \frac{\frac{\Phi(\Gamma_{i+1}) - \Phi(\Gamma_i)}{\$19.7 \times 10^9} + \frac{\Psi_{Pd}(\Gamma_i) - \Psi_{Pd}(\Gamma_{i+1})}{1} + \frac{T_{tr}(\Gamma_{i+1}) - T_{tr}(\Gamma_i)}{106 \text{ min}}}{3} \tag{5.57}$$

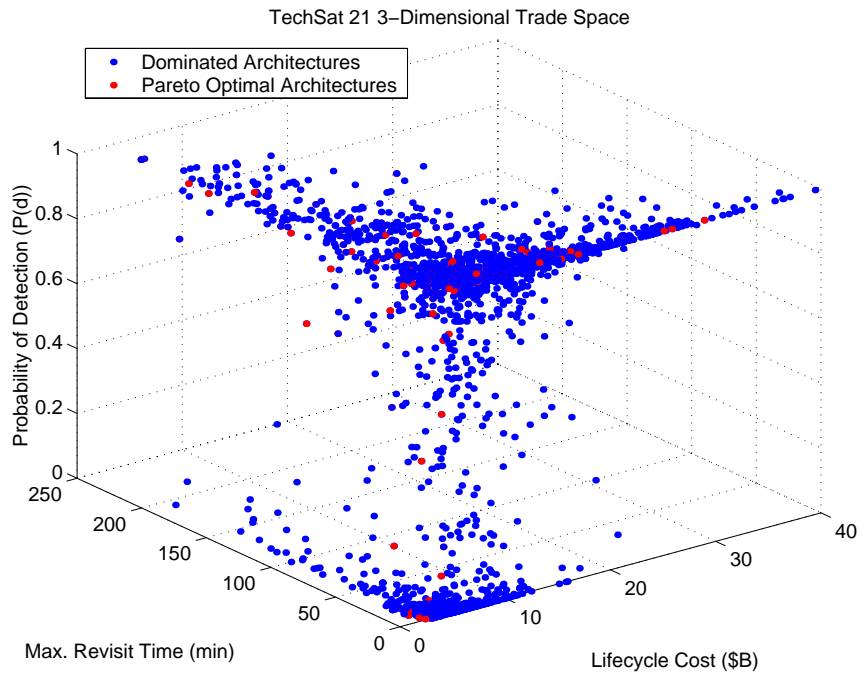
With the correct value for the  $\Delta$  parameter now known from Eqn. 5.57, it is possible to apply the multiobjective multiple solution simulated annealing algorithm to solve Eqn. 5.55. The 2-DOF variant of this MDO algorithm was applied to search for the Pareto optimal system architectures within the TechSat 21 global trade space. Table 5.25 lists the cooling schedule used by the 2-DOF variant of the multiobjective simulated annealing algorithm. The initial system temperature was set to a value of one and reduced by a factor of 0.998 after each iteration. The algorithm was terminated after the completion of 2000 iterations.

**TABLE 5.25** Simulated Annealing Cooling Schedule

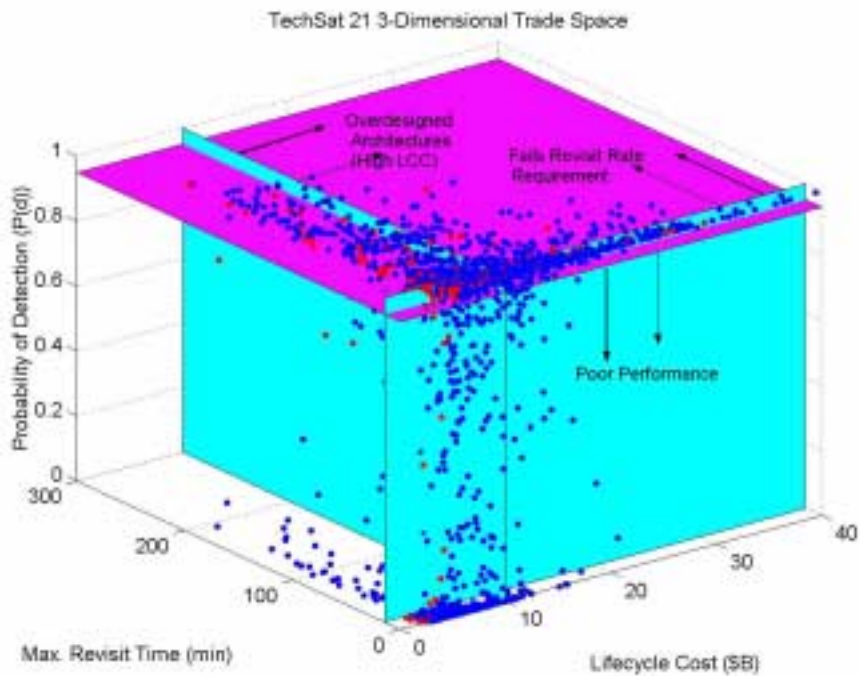
Cooling Schedule Parameters	Value
Initial System Temperature	1
Temperature Reduction Factor	0.998
Number Iterations	2000

## Results

Figure 5.45a illustrates the results from the application of the 2-DOF multiobjective multiple solution simulated annealing algorithm which searches for TechSat 21 architectures that simultaneously minimize lifecycle cost, maximize target probability of detection, and minimize theater revisit times. The red dots represent members of the Pareto optimal set of system architectures found by the algorithm. Once a Pareto optimal set within the global trade space has been found, it may be further partitioned to focus on the region of most interest to the customer. Pareto optimal architectures below the purple plane exhibit unacceptable performance for the AFRL. Pareto optimal architectures to the right of the first blue plane exceed the allocated program budget, while Pareto optimal architectures to the left of the second blue plane fail the 15 minute theater revisit requirement. This leaves the Pareto optimal system architectures in the top front corner of the plot as the conceptual designs of most interest to the TechSat 21 customer. Thus, these are the architectures that would be recommended for further detailed study in the next phase of the TechSat 21 program as they represent the most efficient designs to meet the mission goals. Notice that



(a)



(b)

**Figure 5.45** Results from the Application of the Multiobjective Multiple Solution Simulated Annealing Algorithm (DOF=2) on the TechSat 21 Global Trade Space (a) with the AFRL Requirements Superimposed (b)

this method provides the customer with different options to achieve the same goals. The customer may then base his selection of the final system architecture on further detailed analysis, his own program priorities (i.e. budget, risk tolerance, etc.), a multi-attribute utility theory analysis, or other factors.

Just as a line was fit to the Pareto optimal set of TechSat 21 system architectures in stage three of the test matrix (i.e. optimizing a 2-dimensional objective function), a surface may be fit to the Pareto optimal set found from optimizing a 3-dimensional trade space. This surface represents the boundary of most efficient design solutions along which the systems engineer can't change the design to improve one metric (i.e. one term in the objective function, one design goal, or one decision criteria) without simultaneously hurting at least one other metric. For example, moving from one point on the 3-dimensional Pareto optimal surface to another point may increase the target probability of detection and decrease the maximum theater revisit time, but at the expense of increasing the lifecycle cost.

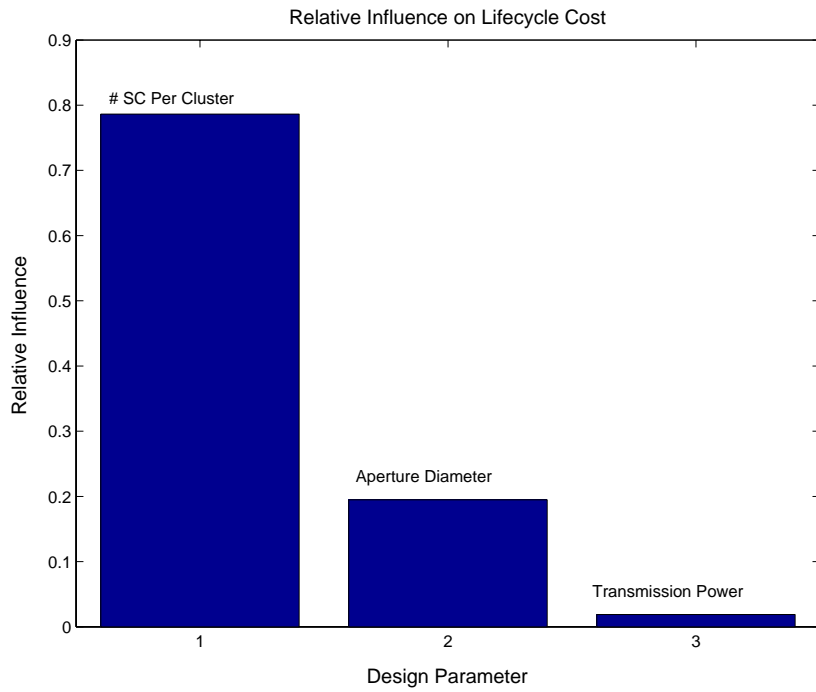
### **Summary**

In this section, the 2-DOF multiobjective multiple solution simulated annealing algorithm has been applied to search the TechSat 21 global trade space for the Pareto optimal set of system architectures to the three dimensional optimization problem in Eqn. 5.55 without complete enumeration of the enormous trade space. This Pareto optimal set contains the most efficient TechSat 21 architectures the AFRL may choose from for the GMTI mission when attempting simultaneously to minimize lifecycle cost, maximize target probability of detection, and minimize the theater revisit time.

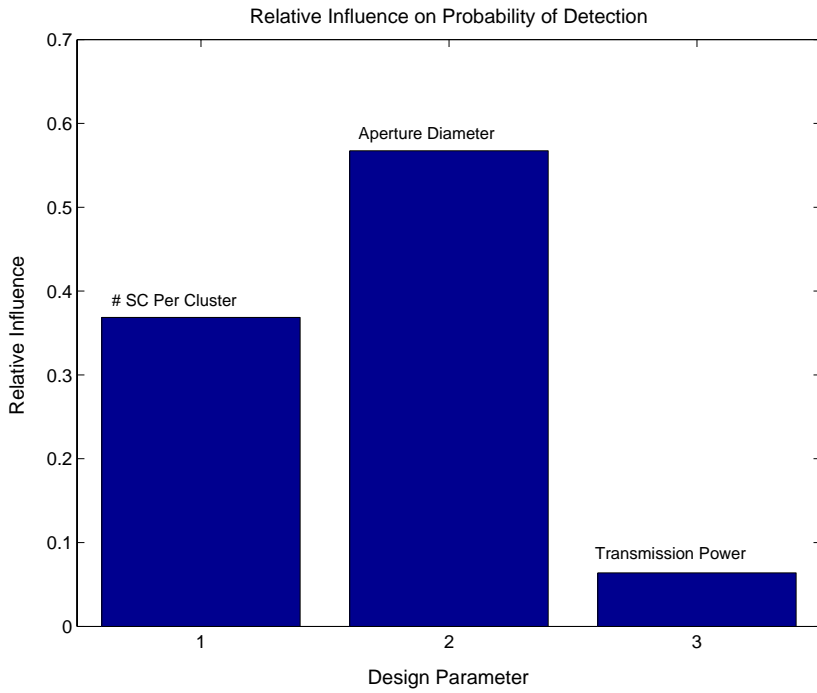
## **5.6 MMDOSA Step 5 – Interpret Results (Sensitivity Analysis)**

Figure 5.46 illustrates the ANOVA results for The Aerospace Corporation CDC TechSat 21 point design (Table 5.7) local trade space (Figure 5.30). As one can see, the number of satellites per cluster exerts by far the greatest relative influence on the system lifecycle cost, while the radar antenna aperture diameter exerts the greatest relative influence on the





(a)



(b)

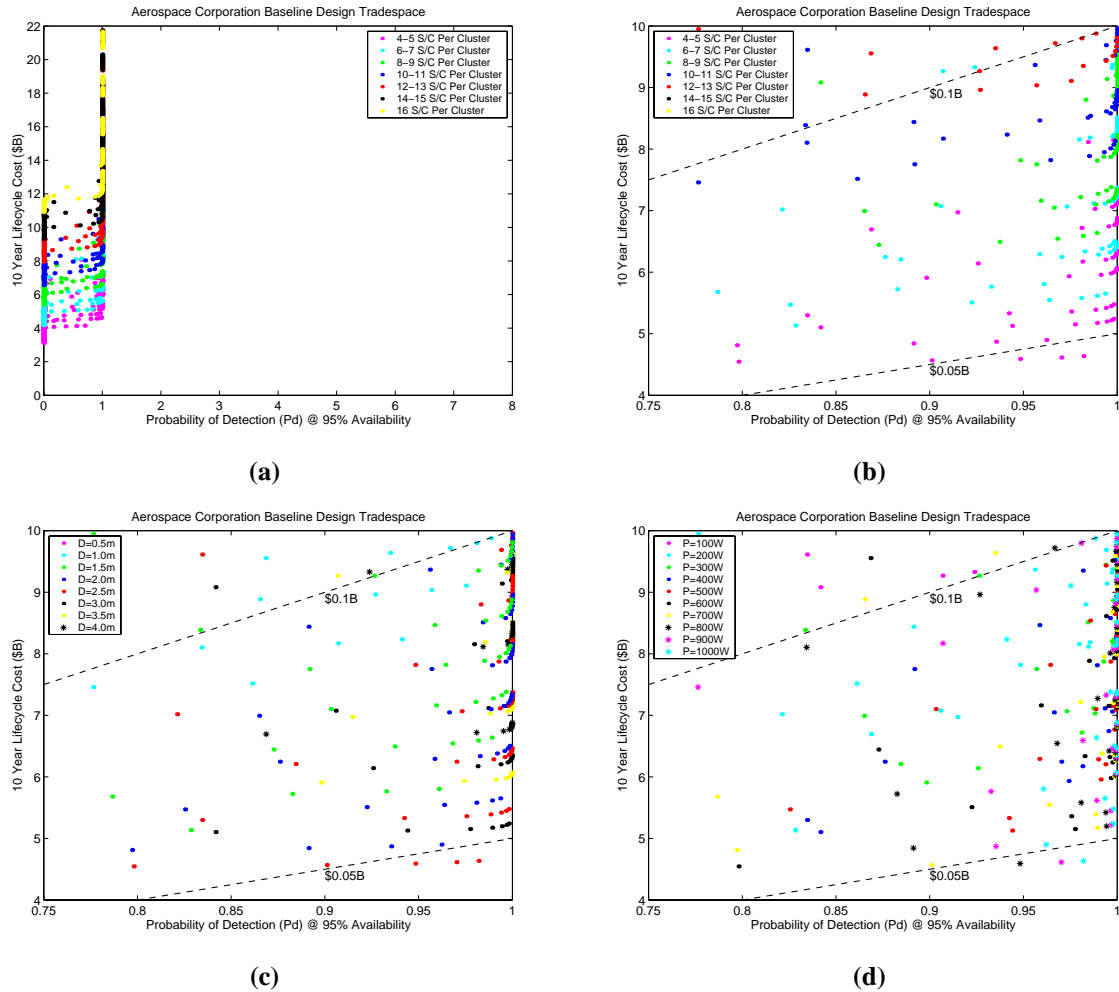
**Figure 5.46** ANOVA Results for The Aerospace Corporation CDC TechSat 21Design Trade Space with Respect to Lifecycle Cost (a) and Target Probability of Detection (b)

target probability of detection. Notice that relative to these two design vector variables, the radar payload power exerts little influence on the design attributes.

ANOVA provides the systems engineer with information on which design parameters give the most leverage over various aspects of the design. The ANOVA results may also be used to guide technology investment decisions. For example, if the TechSat 21 program has a limited technology investment budget and reaching the desired target probability of detection is the foremost priority, then it appears from the ANOVA results that those funds would be best spent on technologies that will yield light-weight and low-cost large-diameter radar transmission antennas. Also, even at the current cost, since the number of spacecraft and aperture size have reciprocal effects of cost and performance, the ANOVA results point to reducing the number of spacecraft (i.e. big cost savings for a small performance loss) and increasing the aperture size (i.e. big performance increase for a small cost increase).

Further insight into the TechSat 21 trade space may be gained simply by viewing the trade space from different perspectives. Figure 5.47 illustrates four different views of the Aerospace Corporation CDC design local trade space illustrated in Figure 5.30. In Figure 5.47a, the design architectures are color-coded according to the number of satellites per cluster. The x-axis has been intentionally stretched out to unreasonable probability of detection values to emphasize the s-curve relationship between the number of satellites per cluster design variable and the system attributes of lifecycle performance and lifecycle cost.

The remaining three plots in Figure 5.47 zoom in to the most cost-effective region of The Aerospace Corporation CDC design local trade space. Figure 5.47b color-codes the design architectures according to the number of satellites per cluster. As one can see, small arcs appear within the trade space. These arcs may be differentiated according to the number of satellites per cluster, with systems having fewer satellites per cluster yielding more cost-effective system architectures. Figure 5.47c color-codes the design archi-



**Figure 5.47** Different Views of The Aerospace Corporation CDC TechSat 21 Design Trade Space

tectures according to the radar TRAM module antenna aperture diameter. Again, small arcs appear. This time the arcs are grouped according to aperture diameter, with the middle values of 2 m to 3 m yielding the most cost-effective families of system architectures. Finally, Figure 5.47d color-codes the system architectures according to the radar payload power. Unlike the previous two cases, the arcs in this plot do not correlate with the color-coded design vector parameter. Rather, it appears that cost-effective architectures for the TechSat 21 GMTI mission may be achieved with any radar payload power above 200 W. In other words, there exists less correlation between radar payload power and the system attributes than between the other design vector variables (i.e. number of satellites per cluster and aperture diameter) and the system attributes. The results correlate with the obser-

vations from Figure 5.46. Now that the TechSat 21 MDO results have been interpreted through a sensitivity analysis, step five of the MMDOSA methodology is complete.

## 5.7 MMDOSA Step 6 – Iterate

The results from the sensitivity analysis in Section 5.6 illustrate the importance of the satellite lifecycle cost and radar performance models. The fidelity of the lifecycle cost model was improved in an iterative manner through collaboration with The Aerospace Corporation and the AFRL, which provided cost data specific to the TechSat 21 program, the TechSat 21 satellite bus, and the TechSat 21 transmit/receive antenna module. Additionally, the single objective simulated annealing algorithm was warm started from the final solution to a previous run of the same algorithm to find even more cost-effective TechSat 21 architectures in the global trade space (Table 5.26).

**TABLE 5.26** Best TechSat 21 Architectures Found within the Global Trade Space After Warm Start

CPF (\$M/Pd)	LCC (\$B)	P(d) (%)	Theater Revisit Time (min.)	# Sats. Per Cluster	Apert. Diam. (m)	Radar Trans. Power (W)	Const. Altitude (km)	# Clusters Per Plane	# Orbital Planes
<b>Single Objective Simulated Annealing Algorithm Warm Start Run</b>									
25.4	2.47	0.969	13	4	3.5	1000	1400	4	3
26.5	2.46	0.926	13	4	3.5	900	1500	4	3
<b>Single Objective Simulated Annealing Algorithm Original Runs</b>									
28.1	2.79	0.994	13	4	4	1000	1500	4	3
28.2	2.60	0.924	14	5	3.0	700	1300	4	3
29.8	2.94	0.987	14	5	3.5	700	1300	4	3
30.6	2.68	0.875	8	4	3.0	900	1200	4	4
33.2	3.14	0.945	14	4	2.5	900	1000	4	6

## 5.8 MMDOSA Step 7 – Converge on Best System Architectures

Having applied the first six steps of the MMDOSA methodology to the TechSat 21 conceptual design problem, the best architectures for the Air Force Research Laboratory to carry forward into Phase B may be recommended. Architectures are recommended for each of the following three design scenarios:

1. The AFRL wishes to restrict itself to a TechSat 21 architecture that adheres to the The Aerospace Corporation CDC constellation geometry.
2. The AFRL wishes to select a TechSat 21 architecture that minimizes the cost per function metric irrespective of the constellation geometry as long as the system adheres to the 15 minute maximum theater revisit time requirement.
3. The AFRL wishes to select a TechSat 21 architecture that simultaneously minimizes lifecycle cost, maximizes target probability of detection, and minimizes theater revisit time.

### 5.8.1 Recommended Architectures Within The Aerospace Corporation Trade Space

Table 5.27 lists the best TechSat 21 architectures within the trade space if the AFRL chooses to restrict itself to systems that adhere to the The Aerospace Corporation CDC constellation geometry. These architectures represent conceptual designs within the Pareto optimal set with a target probability of detection greater than 95%. Within this set of recommended architectures, the systems engineer cannot improve system performance without increasing lifecycle cost. This set of recommended architectures also contains the minimum CPF architecture (i.e. CPF=\$36.9M) within the Aerospace Corporation baseline design local trade space.

**TABLE 5.27** Best Architectures Within the TechSat 21 Aerospace Corporation Baseline Design Trade Space

Architecture	1	2	3	4	5	6	7	8	9
# Satellites Per Cluster	4	4	4	4	4	4	4	4	4
Aperture Diameter (m)	2.0	2.0	2.5	2.5	2.5	2.5	2.5	3.0	3.0
Radar Transmission Power (W)	900	1000	600	700	800	900	1000	700	800
Max. Theater Revisit Time (min.)	15	15	15	15	15	15	15	15	15
Target Probability of Detection	0.953	0.964	0.974	0.986	0.992	0.996	0.997	0.998	0.999
Lifecycle Cost (\$B)	3.54	3.56	4.02	4.04	4.06	4.08	4.10	4.55	4.57
Cost Per Function (\$M/Pd)	37.1	36.9	41.3	41.0	41.0	41.0	41.1	45.6	45.7

Closer inspection of the recommended architectures reveals distinct patterns. First, all the recommended architectures contain only four satellites per cluster. Minimizing the number of satellites per cluster reduces lifecycle cost and, according to the deconvolution sig-

nal processing algorithm model used in the radar module, four satellites can provide adequate target probability of detection provided the radar power-aperture product is large enough at the given altitude. The aperture diameter ranges between 2.0 m - 3.0 m. Smaller diameters fail to provide enough power-aperture to detect targets reliably and larger apertures drive up the lifecycle cost too high. Finally, the radar payload power ranges between 600 W and 1000 W. Increasing the transmission power exerts a negligible effect within the uncertainty of the cost models on the lifecycle cost of the system, with significant increases in lifecycle cost of approximately \$500M occurring with each 0.5 m increase in aperture diameter. Based on these observations, Table 5.28 lists the family of architectures the MMDOSA analysis recommends to be carried forward into the next phase of the TechSat 21 design process if the AFRL chooses to restrict itself to systems that adhere to the The Aerospace Corporation constellation geometry.

**TABLE 5.28** Recommended Family of Architectures for Phase B

<b>Design Vector</b>	<b>Value</b>
# Satellites Per Cluster	4
Aperture Diameter	2.0 m - 3.0 m
Radar Transmission Power	1000 W
Constellation Altitude	800
# Clusters Per Orbital Plane	5
# Orbital Planes	7

## 5.8.2 Recommended Architectures within the Global Trade Space

### Best Single Objective Optimization Architectures

Table 5.26 lists the best TechSat 21 architectures found by the single objective simulated annealing algorithm within the global trade space of 732,160 architectures. By allowing the constellation geometry parameters within the design vector to vary, more cost-effective TechSat 21 architectures were found than those present within The Aerospace Corporation CDC baseline design trade space. The total number of clusters in the constellation, and thus the lifecycle cost of the system, was reduced while still meeting the 15 minute

theater revisit time requirement by placing the constellation at a higher altitude. Increasing the altitude, however, necessitates an increase in aperture diameter to provide enough power-aperture to overcome the increased signal path loss that occurs due to the increased range to target at higher altitudes. It turns out that the decrease in lifecycle cost that results from using fewer clusters (i.e. building fewer satellites) is greater than the increase in the unit costs of building and launching larger satellites to support larger antennas. These trends lead to the recommendation of the family of architectures in Table 5.29 to be carried forward into Phase B of the TechSat 21 program if the AFRL wishes to select a TechSat 21 architecture that minimizes the cost per function metric irrespective of the constellation geometry as long as the system adheres to the 15 minute maximum theater revisit time requirement. It should be noted that placing the satellites at these higher orbital altitudes will increase their radiation exposure, and this issue should be studied in greater detail during Phase B.

**TABLE 5.29** Recommended Family of Architectures for Phase B

<b>Design Vector</b>	<b>Value</b>
# Satellites Per Cluster	4 - 5
Aperture Diameter	3.0 m - 4.0 m
Radar Transmission Power	1000 W
Constellation Altitude	1300 km - 1500 km
# Clusters Per Orbital Plane	4
# Orbital Planes	3

### **Best Multiobjective Optimization Architectures**

Table 5.30 lists in order of increasing lifecycle cost the best TechSat 21 architectures found by the multiobjective simulated annealing algorithm within the global trade space of 732,160 architectures. These architectures represent conceptual designs within the Pareto optimal set with a lifecycle cost less than \$5 B, a target probability of detection greater than 95%, and a maximum theater revisit time of less than 15 minutes. Within this set of recommended architectures, the systems engineer cannot simultaneously reduce

lifecycle cost, improve system performance, and decrease the maximum theater revisit time. These are the architectures which should be carried forward into Phase B of the TechSat 21 program if the AFRL wishes to select an architecture that simultaneously minimizes lifecycle cost, maximizes target probability of detection, and minimizes theater revisit time.

**TABLE 5.30** Best Multiobjective Architectures Found Within the TechSat 21 Global Trade Space

Architecture	1	2	3	4	5	6	7	8	9	10	11	12
# Satellites Per Cluster	5	5	5	5	5	9	7	7	7	5	6	6
Aperture Diameter (m)	3.0	3.5	3.0	3.0	3.5	2.5	4.0	2.5	4.0	4.0	3.0	3.0
Radar Transmission Power (W)	900	900	900	900	600	1000	700	700	900	800	800	1000
Constellation Altitude (km)	1500	1300	1100	1300	1500	1400	1500	1300	1400	1200	1300	1400
# Clusters Per Orbital Plane	4	4	5	5	6	4	4	7	4	5	6	6
# of Orbital Planes	4	4	4	4	3	3	3	3	3	4	4	4
Max. Theater Revisit Time (min.)	5	7	11	5	4	13	13	3	13	6	1	0
Target Probability of Detection	.951	.996	.996	.986	.968	.999	.999	.981	.999	.999	.997	.998
Lifecycle Cost (\$B)	3.17	3.59	3.65	3.67	3.80	4.08	4.38	4.39	4.41	4.69	4.85	4.88
Cost Per Function (\$M/Pd)	33.3	36.0	36.7	37.3	39.3	40.9	43.8	44.7	44.1	46.9	48.6	48.9

## 5.9 Conclusions

This chapter has illustrated the application of the MMDOSA methodology to the conceptual design of the TechSat 21 GMTI mission. First, the GINA methodology was applied to create a GINA simulation model for the system. Next, univariate studies and random sampling were executed to obtain initial information on the trade space. With this information in hand, single objective and multiobjective simulated annealing algorithms were applied to search for the most cost-effective architectures and the Pareto optimal architectures within the trade space. After a sensitivity analysis, specific system architectures were identified and recommended for further detailed study in the next phase of the TechSat 21 program.

Table 5.31 compares the existing Aerospace Corporation TechSat 21 CDC point design architecture with the best design architecture found via the MMDOSA methodology under



each of the three design scenarios introduced in Section 5.8. In the first design scenario, the MMDOSA methodology identified a system architecture within The Aerospace Corporation baseline design local trade space that reduces the CPF metric by 49%. In the second design scenario, the MMDOSA methodology identified a system architecture within the global trade space that reduces the CPF metric by 65%. In the third design scenario, the MMDOSA methodology identified a system architecture within the global trade space that reduces the lifecycle cost by 57% and the maximum theater revisit time by 67% at the expense of a 5% drop in target probability of detection. As one can see, in each instance the MMDOSA methodology enabled the systems engineer to identify better, more cost-effective architectures than any previously considered. In this manner, the MMDOSA methodology improves the quality of the work performed by the systems engineer during the conceptual design phase of a military DSS program.

**TABLE 5.31** Comparison of Existing Designs vs. Design Found by the MMDOSA Methodology

Architecture	CPF (\$M/Pd)	Lifecycle Cost (\$B)	Target Probability of Detection	Maximum Revisit Time (min)
<b>Aerospace Corp. Trade Space</b>				
Aerospace Corp. Design	72.9	7.29	0.999	15
MMDOSA Design	36.9	3.56	0.964	15
% Improvement	49%	51%	-4%	0%
<b>Global Trade Space</b>				
Aerospace Corp. Design	72.9	7.29	0.999	15
MMDOSA Single Objective Design	25.4	2.47	0.969	13
% Improvement	65%	66%	-3%	13%
<b>Global Trade Space</b>				
Aerospace Corp. Design	72.9	7.29	0.999	15
MMDOSA Multiobjective Design	33.3	3.17	0.951	5
% Improvement	54%	57%	-5%	67%



# Chapter 6

## **BROADBAND COMMUNICATIONS MISSION CASE STUDY AND RESULTS**

Chapter 6 presents the third class of space missions to which MMDOSA may be applied – commercial missions. Both single objective and multiobjective MDO algorithms are applied to a broadband communications mission to search for the most cost-effective architectures within the trade space. The commercial case study presented here differs from the civil and military case studies presented in Chapter 4 and Chapter 5 in several important ways. First, lifecycle revenues are taken into account along with lifecycle costs when computing the relative economic competitiveness of alternative system architectures. Second, the multiobjective version of the broadband conceptual design optimization problem contains more decision criteria than the equivalent TPF and TechSat 21 multiobjective optimizations, and thus illustrates the ability of MMDOSA to find higher-order Pareto optimal sets. This chapter demonstrates the step-by-step application of the MMDOSA methodology to the conceptual design of the broadband mission and concludes by illustrating how MMDOSA found better system architectures during the conceptual design phase than the existing point design architectures for this mission.

### **6.1 Introduction to the Broadband Mission**

Broadband systems are high capacity networks that offer high data rate connections on the order of megabits per second (Mbps). In contrast, dial-up telephone modems transfer data at only 56 kbps. Due to the growth of the internet and other information transfer services

in today's global economy, the demand for broadband networks is expected to grow rapidly in the near future. Potential key markets identified for broadband services include:

- High speed internet connections for individual consumers,
- High speed intranet/extranet connectivity for businesses,
- Interactive TV,
- Fixed telephony in underdeveloped regions, and
- The next generation Air Traffic Control System.

It is expected by industry analysts that wherever terrestrial broadband networks (i.e. fiber optic cable and DSL) already exist, satellite broadband networks will not be cost-effective [Microcom Systems, 2002]. However, there still exist large regions of the developing world that are not expected to attain terrestrial broadband access for several decades. These regions represent a key potential market for broadband satellite systems. One market study estimates the global market value for broadband services to surpass \$1.5 trillion annually by the year 2010 [Motorola, 1997]. Even if satellite systems only capture 10% of this market, these systems will cumulatively earn \$150 billion per year. It should be noted that with the recent global economic slowdown, the broadband market has not grown as rapidly as predicted, and the above market estimate may be liberal. Still, this tremendous earnings potential has led to the proposal of different satellite broadband networks by many well known aerospace and telecommunications companies, several of which have already filed license applications with the U.S. Federal Communications Commission (FCC) for their proposed systems. The remainder of this chapter details how MMDOSA has been applied to the conceptual design of the broadband satellite communications mission.

## **6.2 MMDOSA Step 1 – Create the GINA Model**

GINA – the Generalized Information Network Analysis methodology – is a systems engineering and architecting framework for the comparative analysis of distributed satellite systems [Shaw et al, 2001]. GINA enables the comparison of different system architec-

tures for a given mission. The foundation behind the GINA methodology is the assertion that most satellite systems are information disseminators that can be represented as information transfer networks [Shaw et al, 2000]. A summary of the steps in the GINA methodology is presented in Section 3.2. Through these steps, GINA allows the systems engineer to make meaningful, quantitative trades at the conceptual design level by directly relating lifecycle performance to lifecycle cost. The remainder of this section details the application of the GINA methodology to the broadband communications mission.

### **6.2.1 GINA Step 1 – Definition of the Mission and CDP Objectives**

In defining the mission objective, the systems engineer must explicitly state what the real application of the system is in terms of the user needs. To do this, the customer of the system must first be identified. In the case of the broadband mission, there exist two separate customers. The first is the commercial entity procuring the system with the intent to use it to generate a profitable business. This customer would ideally prefer a system that generates the greatest amount of revenue for the lowest initial investment. The second customer is the end-user of the broadband service. This second customer would prefer a system that provides a high quality service for a low unit price. The conflict between these two opposing communities will be captured and illustrated in the GINA model.

Once the customer has been identified, the mission objective may be defined. The broadband mission objective is:

To deliver broadband service for the commercial marketplace with a system that maximizes shareholder earnings and value.

Close inspection of this statement reveals what the system must accomplish to meet the needs of the user. First, the system must provide broadband service – that is high data rate delivery at 1.54 Mbps (i.e. a T1-minute) – to end user terminals. This will require the use of Ka and Ku band frequencies. Second, in order to maximize shareholder earnings and value, the system must be designed to be profitable – that is to generate large revenues for as small an initial investment as possible. Thus, defining the mission objective in terms of

the user needs begins to elucidate the form and function of the space system to be designed. It also provides the top-level requirements of the system. In the case of the broadband mission, this equates to being able to deliver a high data rate (i.e. 1.54 Mbps) at a bit error rate of  $10^{-5}$ . For comparison, high-bandwidth terrestrial digital subscriber line (DSL) services deliver data at rates ranging from 512 Kbps to 6.1 Mbps and a bit error rate of  $10^{-6}$  to  $10^{-9}$ .

Next, the systems engineer must explicitly define the objective to be achieved during the conceptual design phase to remove all ambiguity concerning what will have been accomplished by the end of this phase of the design process. For the broadband case study, the CDP objective was:

To develop a framework for the comparison of broadband satellite constellations spanning the system trade space, implement the framework as a software tool, and use the tool to identify broadband satellite system architectures that provide the greatest shareholder earnings and value.

It is important to stress that the conceptual design phase goal is *not* to create a single point design, but rather to create a tool that may be used to model, analyze, and compare different system architectures for the broadband mission.

### **6.2.2 GINA Step 2 – Information Network Representation**

Modeling the broadband system as an information network entails defining all of the origin-destination pairs in the system (Figure 6.1). The origin is the source of the information to be transmitted. Potential sources include broadcasting centers, commercial entities such as banks, and internet servers. The sink is the end-user terminal receiving the information, and might include individual internet subscribers, corporate users (i.e. banks to check credit records in real time), and government users (i.e. the FAA to facilitate air traffic control).

In the GINA methodology, the capability of an architecture is characterized by four quality of service parameters that relate to the detection process, and to the quantity, quality, and availability of the information that is processed through the network. These four

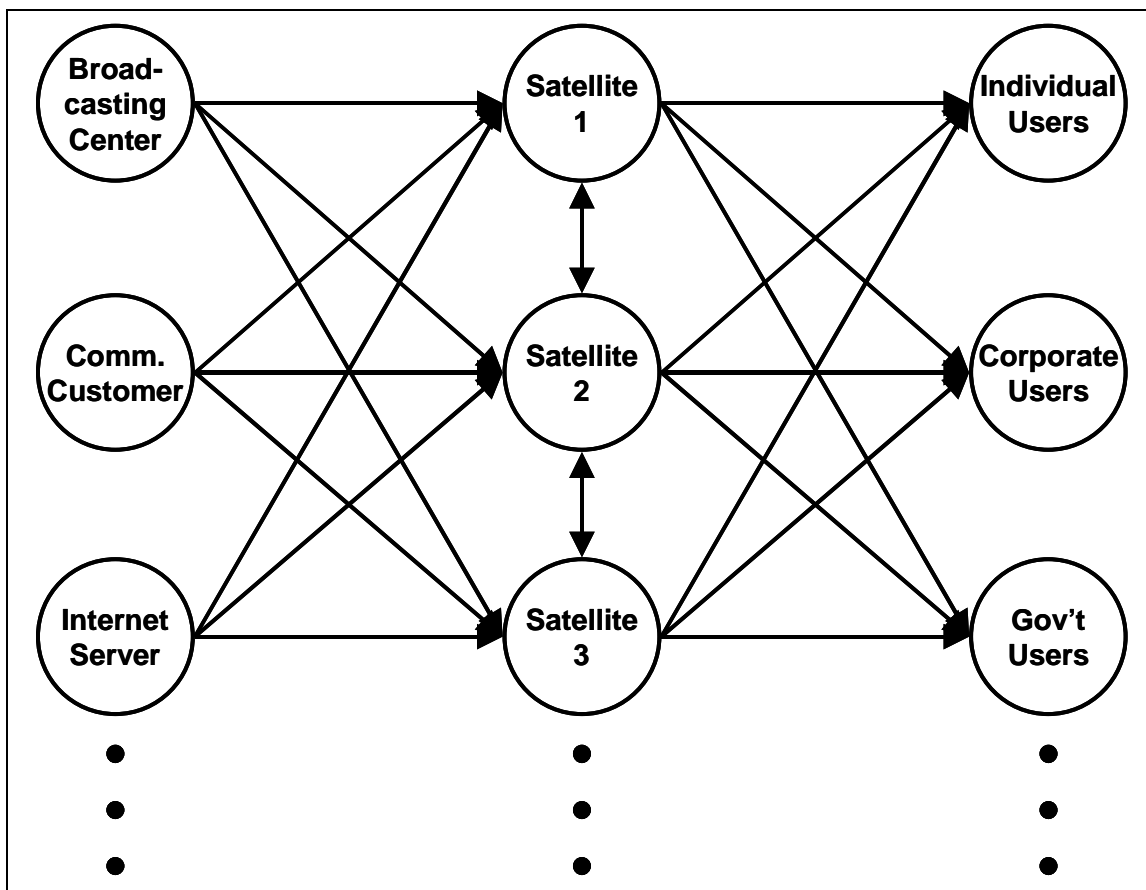


Figure 6.1 The Broadband System Modeled as a Network

parameters are signal isolation, information rate, information integrity, and the availability of these services over time [Shaw et al, 2000]. Once formulated, these four metrics serve as the minimum instantaneous capability requirements the system must meet to satisfy the customer. These capability metrics assess how well the aforementioned customer requirements are met at an instantaneous point in time.

- Isolation  
Isolation refers to the ability of a system to isolate and distinguish information signals from different signal and noise sources within the network. For the broadband mission, the access scheme and the signal-to-noise ratio define the isolation metric.
- Integrity

Integrity is a measure of the quality of the information being transferred through the network. In the broadband mission, this is quantified by the bit error rate (BER).

- Rate

Rate measures the speed at which the system transfers information between the nodes (i.e. sources and sinks or O-D pairs) in the network. For the broadband mission, this is measured in bits per second.

- Availability

Availability characterizes the instantaneous probability that information is being transferred through the network between all of the sources and sinks (i.e. all the nodes). For the broadband mission, this is the percentage of time that the system can provide a T1-link with a BER no worse than  $10^{-5}$  to a random user terminal.

At this point, the systems engineer must decide which of these capability parameters will become hard requirements that the system must meet to be considered feasible and which parameters will serve to distinguish between system architectures. For the broadband mission, the isolation, integrity, and availability parameters are hard requirements every potential architecture must meet, while the cumulative rate will differ between architectures. The higher the cumulative rate the system can provide, the greater the number of simultaneous T1-connections the system can support, and therefore the greater revenue the system can generate. Table 6.1 summarizes the capability parameters as applied to the broadband mission along with the requirements placed on these parameters.

**TABLE 6.1** The GINA Capability Quality of Service Metrics for the Broadband Mission

Capability	Equivalent Design Parameter	Requirement
Isolation	Access Scheme	FDMA/TDMA
	Number of Channels	1.56 Mbps/Channel
	Signal-to-Noise Ratio	6.4 dB
	Link Margin	7.8 dB
Integrity	Bit Error Rate	$10^{-5}$
Rate	Instantaneous Bits Per Second	Free Variable (Maximize)
Availability	% Time Meet Requirements	95%



### 6.2.3 GINA Step 3 – Develop System Metrics

To compare all the different architectures that are formulated during the conceptual design phase, GINA uses a quantifiable set of metrics – capability, performance, lifecycle cost, cost per function, and adaptability. The capability metrics have already been introduced in Section 6.2.2. The remaining four metrics are discussed below. These are the metrics that will be used to evaluate and compare alternative broadband architectures.

#### **Performance**

While the four capability quality of service parameters measure how well the architecture meets the capability requirements at any instantaneous point in time, the performance metric measures how well the architecture satisfies the demands of the market over the entire life of the mission [Shaw, 1998]. Because the instantaneous maximum data rate was identified as the distinguishing capability metric between broadband architectures, performance is expressed as the total amount of data (i.e. the total number of bits or T1-minutes) that the system delivers over the mission duration. In other words, the performance is the total throughput or capacity of the system.

To calculate this quantity, however, the additional complication of taking into account all possible failures that may occur within the system must be addressed. As individual components within the system fail over time, the instantaneous data delivery rate of the system will decrease in order to maintain the same pre-failure level of integrity (i.e. a  $10^{-5}$  BER). Thus, the instantaneous throughput rate capability metric is a dynamic quantity that changes over time.

To take into account potential failures and the effect they have on system performance, a reliability model is needed. GINA uses Markov reliability modeling techniques to determine both the probability that the system will continue to function over a given amount of time and the likelihood with which the system will function in different partially failed states throughout the mission. A Markov model must be created for each proposed architecture. From the Markov model, a system of differential equations is developed, as

shown in Eqn. 6.1, where  $\dot{\vec{P}}$  is the time rate of change of the state probability vector,  $A$  is the state coefficient matrix comprised of component failure rates, and  $\vec{P}$  is the state probability vector.

$$\dot{\vec{P}} = A\vec{P} \quad (6.1)$$

The solution to this set of linear, first-order, partially coupled ordinary differential equations determines the probability of the broadband system being in any given state at a particular time. Coupling the outputs of the reliability model with the outputs of the capability model (i.e. the instantaneous maximum data delivery rate of the broadband architecture in each operational state) yields the total performance (i.e. cumulative amount of data delivered) of the system. The coupling equation is the utility function:

$$\Psi(T) = \int_0^T \sum_{i=1}^n C_i P_i(t) dt \quad (6.2)$$

where  $\Psi$  is the total amount of data (i.e. bits or T1-minutes) delivered,  $T$  is the mission lifetime,  $n$  is the total number of operational states (ex. state 1 = all of the satellites in the constellation are functional, state 2 = all but one of the satellites in the constellation are functional, etc.),  $C_i$  is the capability (i.e. T1-minutes delivered per day) in each state  $i$ , and  $P_i(t)$  is the probability of being in each operational state  $i$  as a function of time  $t$ .

### Lifecycle Cost

Four primary components of the lifecycle cost model are the payload cost, satellite bus cost, launch cost, and operations cost. Payload and satellite bus theoretical first unit (TFU) costs are computed via cost estimation relationships (CER's) based upon past telecommunications satellites. A learning curve is then applied to compute the total cost in constant year dollars as a function of the required number of satellites for a particular system architecture. The launch cost model uses the optimization algorithm in Appendix B to select a suite of launch vehicles from the complete set being considered to deploy the sys-

tem at minimum cost. Inputs to the launch cost model are the total number of satellites in the architecture, the mass and volume of the spacecraft, the orbital parameters of the constellation, and the performance parameters of each potential launch vehicle. The cost of insuring each launch is also added to the launch costs. Finally, the operations cost model estimates the operations crew size, operations crew cost, flight/ground software size, software costs, terminal costs, and total operations lifecycle cost of operating the broadband system over the desired mission life.

### **Cost Per Function**

The cost per function (CPF) metric provides a clear measure of the cost of an architecture versus its performance [Shaw et al, 2001]. For the broadband mission, the cost per function is defined as the cost per billable T1-minute. In addition to these four components of lifecycle cost, a revenue analysis is also performed on the system. Based upon the total performance of the system, lifecycle cost of the system, desired internal rate of return, expected inflation rate, and assumed market demand; a net present value analysis is performed to compute the price that must be charged per T1-minute of service to achieve a zero net present value over the life of the mission. The lower the cost per T1-minute, the greater the economic competitiveness of the architecture. The additional aspect of calculating revenues distinguishes this commercial case study from the previous civil and military case studies in which the CPF was calculated simply by dividing lifecycle cost by lifecycle performance.

### **Adaptability**

Adaptability measures how flexible an architecture is to changes in design assumptions and mission requirements [Saleh, 2002]. In one sense, adaptability may be thought of as the sensitivity or elasticity of the CPF of a particular architecture to incremental changes in an assumption or requirement. For the broadband mission, potential assumptions that could be altered to measure architecture sensitivity include component costs, learning curve slopes, and market penetration. In another sense, adaptability may be thought of as

the flexibility of a particular architecture to a new set of mission requirements. An example of flexibility for a broadband system might be the ability of an architecture to transition from a commercial data delivery market to a military secure channel, jam-proof data delivery market.

#### **6.2.4 GINA Step 4 – Partition the Conceptual Design Problem**

The next step in the GINA process involves listing all the design variables the systems engineer has control over. From this list, key independent variables that drive the design of the system are extracted and inserted into the design vector. The design vector contains the attributes that will distinguish and differentiate alternative system architectures. Remaining variables are placed in the constants vector. The constants vector contains attributes that the systems engineer has control over, but that will not distinguish or differentiate alternative architectures. For example, the assumed areal density of the satellite solar arrays is set to a constant value as a function of the type of solar array (ex.  $1 \text{ kg/m}^2$  for gallium arsenide multi-junction cells) regardless of the other attributes of the architecture.

For the broadband mission, six variables – the constellation altitude, constellation inclination, number of satellites per plane, number of orbital planes, payload power, and phased array antenna transmission area – make up the design vector. Table 6.2 lists the range of values considered for each design vector variable in the broadband case study. Each design vector defines a unique broadband system architecture.

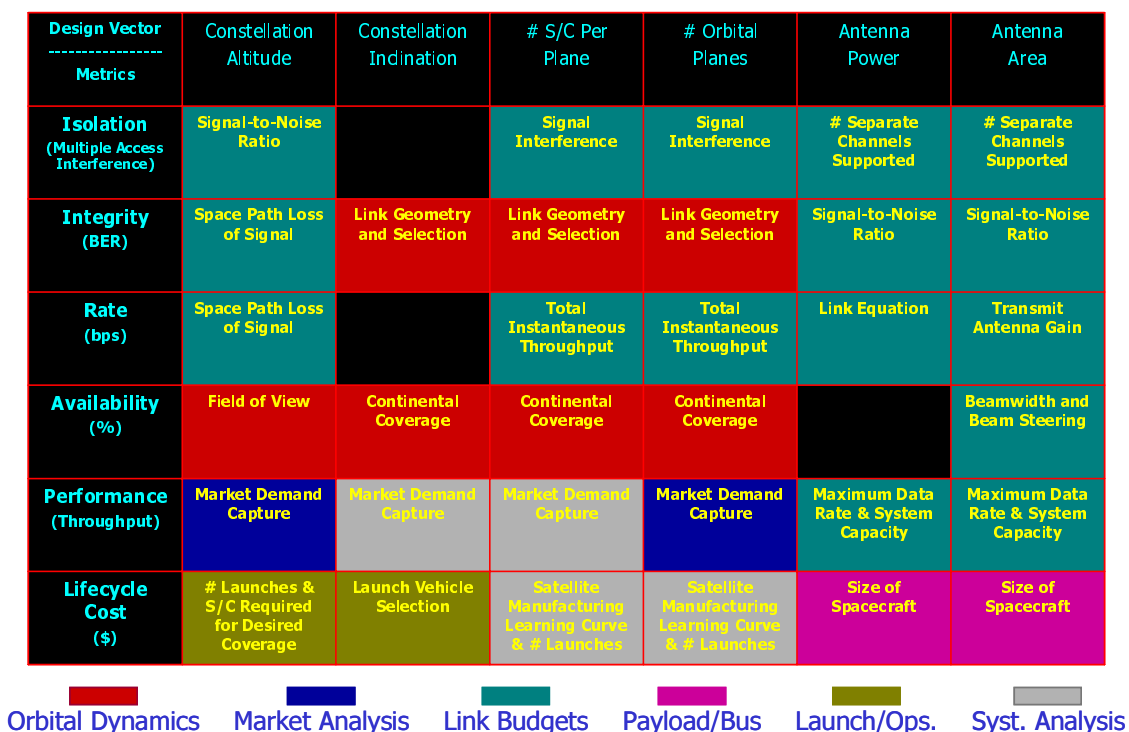
Once the design vector has been identified, it may be matrixed against the GINA system metrics. The chart in Figure 6.2 establishes the relationship between the broadband design vector and the metrics by which competing architectures are judged. It contains the attributes that distinguish individual architectures. Notice the fundamental relationships between the elements of the design vector and the GINA metrics. For example, the payload power directly affects the maximum instantaneous throughput of the system accord-

**TABLE 6.2** The Broadband Design Vector

Design Vector Variable	$\Gamma$	Range	Discretization
Constellation Altitude	$\gamma_1$	LEO,MEO,GEO	Unique
Constellation Inclination	$\gamma_2$	0° - 75°	15°
# Satellites Per Plane	$\gamma_3$	1 - 8	1
# Orbital Planes	$\gamma_4$	1*, 2 - 10	2
Payload Power	$\gamma_5$	1 kW - 10 kW	1 kW
Antenna Transmission Area	$\gamma_6$	0.5 m <sup>2</sup> - 5.0 m <sup>2</sup>	0.5 m <sup>2</sup>
<b>Total # Permutations</b>		<b>42,400</b>	

\*When the inclination is 0° (i.e. equatorial), there can exist only 1 orbital plane in the constellation.

ing to the link equation, but at the expense of satellite mass and cost. Hence, the payload power drives the rate, performance, and lifecycle cost metrics.



**Figure 6.2** Broadband Model Component Identification Metrics Matrix

Once this matrix has been completed, different attributes can be lumped into groups of modeling needs that allow the recognition of important differences between competing architectures. The systems engineer may now define the modules that will be required to develop, analyze, and compare competing architectures for the broadband mission. These modules capture the fundamental physics, costs, and systems engineering trades of the broadband conceptual design problem. Six core modules are required:

1. Orbital dynamics module
2. Market analysis module
3. Link budget module
4. Payload and spacecraft bus module
5. Launch and operations module
6. Systems analysis module

Detailed analysis of the model component identification metrics matrix begins to yield insight to the trade space. Figure 6.3 illustrates an alternative view of the TPF trade space. Each blue arrow represents an element of the design vector, while the green and red boxes represent the benefit in terms of the GINA metrics of increasing or decreasing the value of that parameter. For example, increasing the constellation altitude improves the link geometry, provides for a larger field of view, and enables a greater market capture; but at the expense of increased space path loss, lower signal-to-noise ratio, and higher launch costs. Likewise, decreasing the antenna area reduces the payload costs, satellite bus costs, and launch costs; but at the expense of the signal-to-noise ratio and system throughput.

What makes the broadband conceptual design problem so difficult is that all of these trades are coupled in a nonlinear fashion. One cannot simply optimize each element of the design vector in isolation from each other, because each element of the design vector impacts all elements of the design. This is where the true strength of the MMDOSA methodology demonstrates itself, as MMDOSA enables the systems engineer to capture quantitatively all these couplings and then search the nonconvex trade space for the most cost-effective families of system architectures.

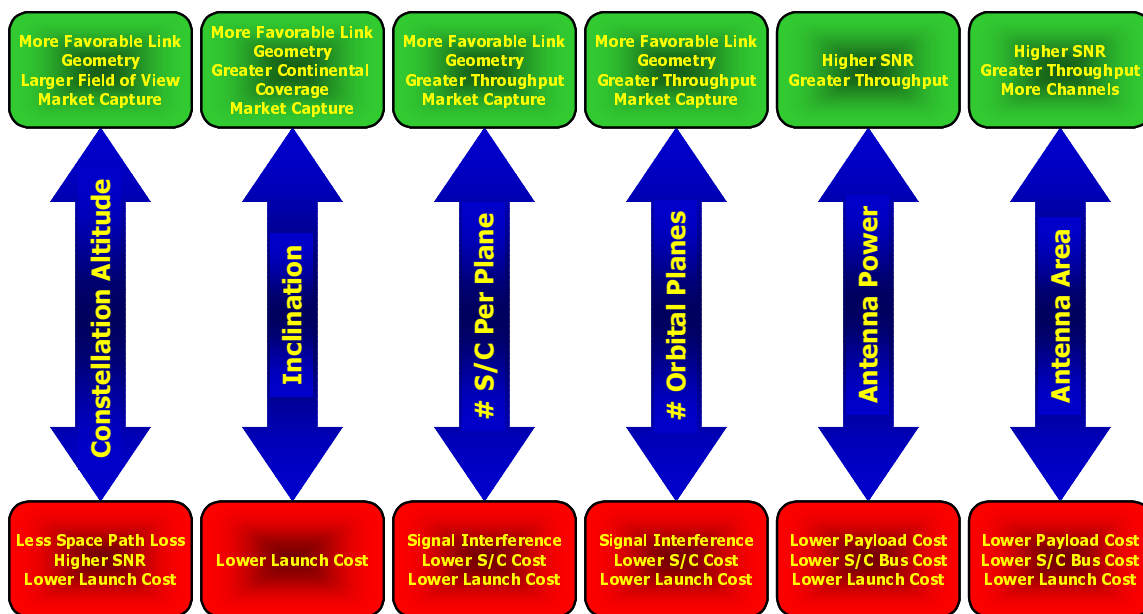


Figure 6.3 Alternative View of the Broadband Trade Space

### 6.2.5 GINA Step 5 – Develop Simulation Software

This section provides a high-level overview of the broadband GINA model as excerpted from the references by Kashitani [Kashitani, 2002]. For greater detail on the broadband GINA model, the assumptions behind the model, and the analytical relationships inside the model; please see these references.

A module is a Matlab m-file that models a specific subsystem or function within the mission. The purpose of the modules is to develop a systematic tool that sizes the subsystems within an architecture for a given set of inputs. If implemented correctly, the modules should also aid in the systems analysis methodology by computing the system metrics for a given architecture. This allows for rapid evaluations and comparisons of alternative broadband architectures.

Figure 6.4 illustrates the Matlab modules identified in the model component identification metrics matrix within the context of the broadband GINA model. The two sets of inputs to the model are the design vector and constants vector. First, the orbital dynamics module computes the access statistics (i.e. time and geometry) between every satellite and every

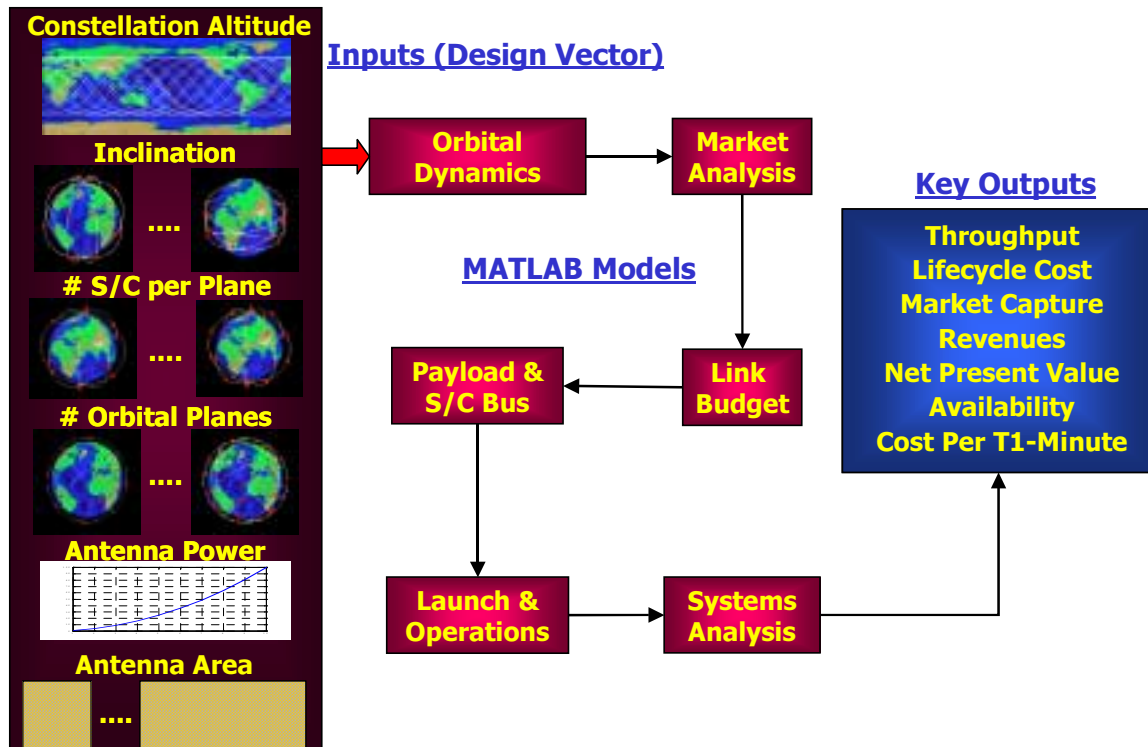


Figure 6.4 Broadband GINA Model Simulation Software Block Diagram

Earth grid point for every time step in the simulation. Next, the market analysis module superimposes a per capita gross national product purchasing power parity world map with a population density map to estimate the market distribution within view of each satellite. The link budget module then performs a link analysis to compute the maximum data rate that can be supported by each satellite for the market within view. The payload and spacecraft bus module sizes and costs the satellites. Next, the launch and operations module uses the optimization approach presented in Appendix B to select a suite of launch vehicles to deploy the constellation, and then estimates the lifecycle launch and operations costs of the system. Finally, the systems analysis module estimates the total throughput of the system over the life of the mission, calculates the lifecycle cost, and performs a net present value analysis to determine the required billing rate for the broadband service to achieve a zero net present value over the life of the mission (i.e. the cost per function).



## Orbital Dynamics Module

Figure 6.5 illustrates key inputs and outputs of the orbital dynamics module. With knowledge of the constellation geometry from the design vector and the minimum elevation angle required to establish a link between a satellite and user terminal from the constants vector, the user terminal access statistics may be computed. Key outputs include the in-view user terminal access locations, times, ranges, and viewing geometries.

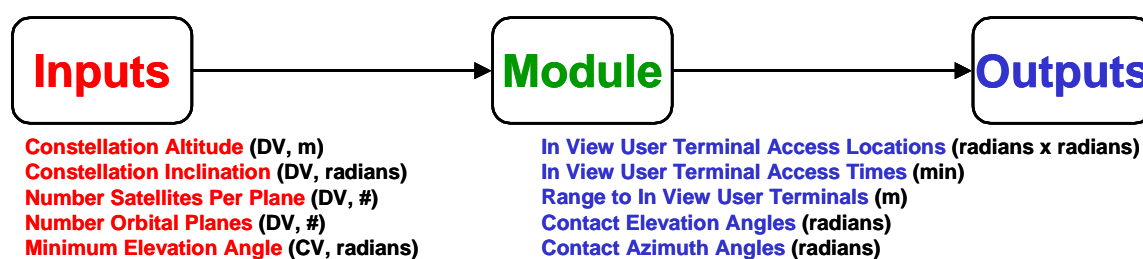


Figure 6.5 Orbital Dynamics Module Inputs and Outputs

A grid is placed over the surface of the Earth at a resolution defined in the constants vector, and a user terminal is placed in the center of each grid point. A J4 orbital propagator that takes into account the non-spherical shape of the Earth then propagates all of the satellites over 24 hours in three minute time steps. At each time step, the geometry between every satellite in the constellation and every user terminal on the Earth grid is analyzed to determine whether or not a T1 link may be established. In this manner, a matrix containing all T1 link access locations, times, and geometries is constructed for later use by the link budget and systems analysis modules.

## Market Analysis Module

Figure 6.6 illustrates key inputs and outputs of the market analysis module. With knowledge of the global broadband market statistics, the fraction of this total market that is willing to subscribe to a satellite service, and the user terminal access locations; the total number of potential subscribers for the satellite broadband service may be estimated. The key output of this module is a market distribution map containing the number of potential subscribers in view of each satellite at each time step in the orbit propagation simulation.

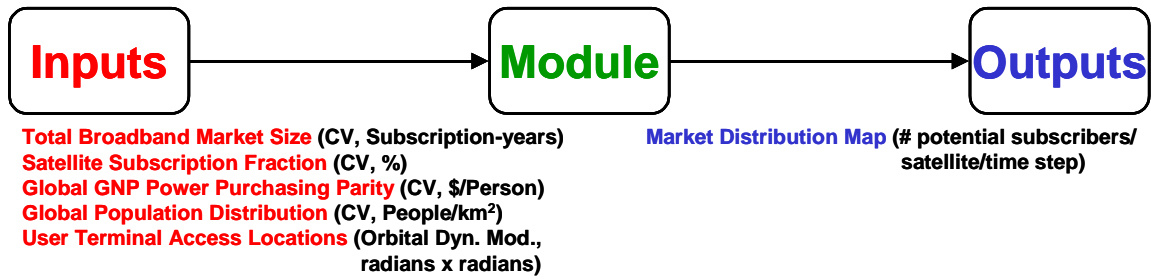


Figure 6.6 Market Analysis Module Inputs and Outputs

Figure 6.7 illustrates schematically how the number of satellite network subscribers is estimated. The outer circle represents the total broadband market which is primarily served by terrestrial broadband networks that typically use fiber optic cable and DSL. Within this total market, it is assumed that a small fraction of the consumer base will be willing to subscribe to a satellite broadband service. The final number of satellite network subscribers will depend upon the geometry of the satellite constellation and the capacity of the satellite network.

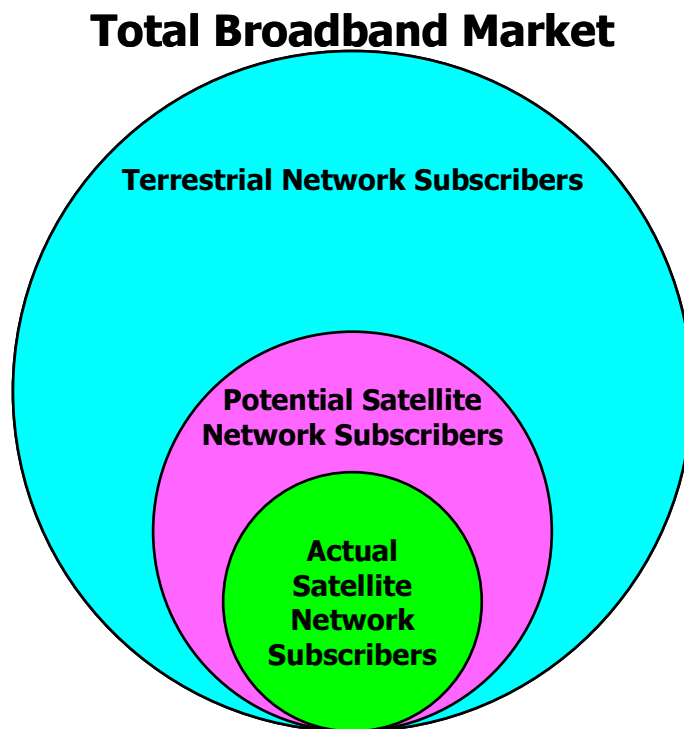


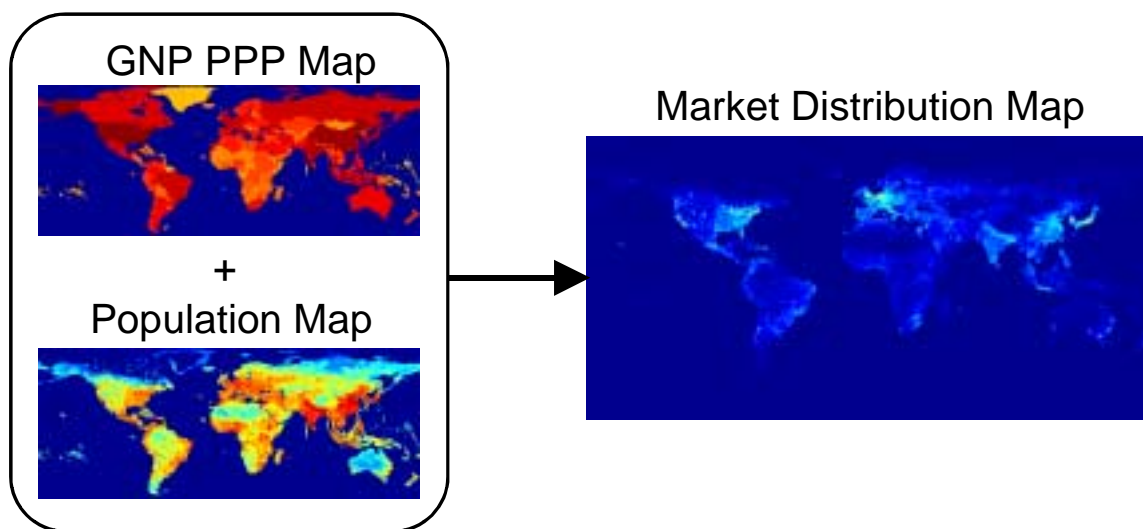
Figure 6.7 Estimating the Potential Number of Satellite Subscribers

Table 6.3 lists the estimated future market size for broadband services according to a report by Pioneer Consulting [Pioneer Consulting, 1998]. This table also lists the number of potential subscribers for a broadband satellite service as a function of maximum possible market penetration. All of the work in this case study assumes a maximum possible market penetration of one percent.

**TABLE 6.3** Potential Satellite Broadband Subscriber Base [Kashitani, 2002]

	<b>Year 2006</b> (millions of subscribers)	<b>Year 2010</b> (millions of subscribers)	<b>Year 2016</b> (millions of subscribers)
<b>Total Market Size</b>	106	252	541
<b>Maximum Possible Market Penetration</b>			
10%	10.6	25.2	54.1
1%	1.06	2.52	5.41
0.1%	0.106	0.252	0.541
0.01%	0.0106	0.0252	0.0541

Figure 6.8 illustrates how the market analysis module computes the actual number of potential subscribers in view of each satellite at each time step. A global gross national product purchasing power parity map is superimposed with a global population density map



**Figure 6.8** Market Distribution Estimation [Kashitani, 2002]

map to develop a global market distribution map. This market distribution map is used downstream in the systems analysis module to determine the number of links each satellite supports during each time step in the simulation.

### Link Budget Module

Figure 6.9 illustrates key inputs and outputs of the link budgets module. With knowledge of the antenna transmission power and area, the geometry between each satellite and user terminal, and other link parameters; the link capacity – the number of simultaneous T1 links – of each satellite during each time step may be computed. The key output of this module is a matrix of link capacities for all of the satellites in the constellation over all of the time steps in the simulation. Table 6.4 lists the assumed values for the different link budget parameters as set in the constants vector.

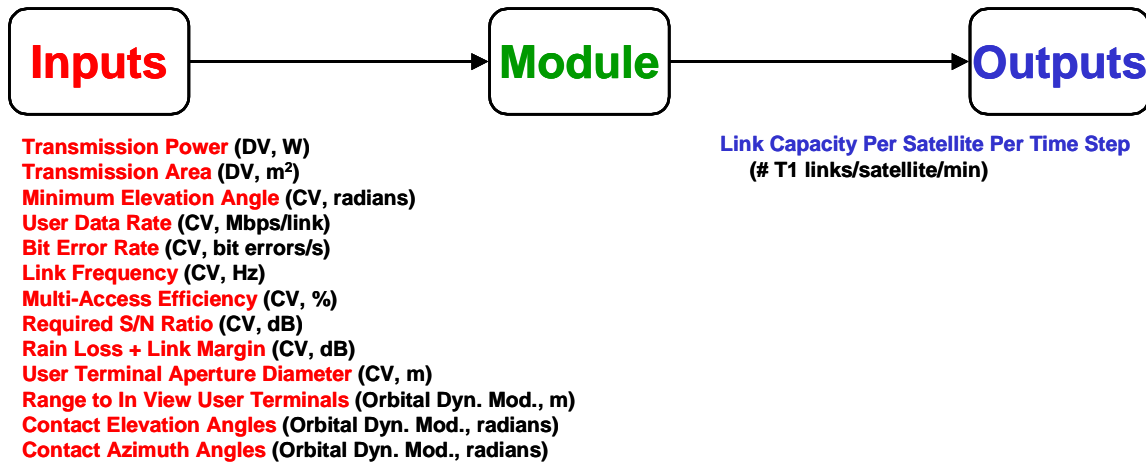


Figure 6.9 Link Budget Module Inputs and Outputs

The path loss  $L_S$  of the transmission signal is computed as:

$$L_S = \left( \frac{\lambda}{4\pi S} \right)^2 \quad (6.3)$$

**TABLE 6.4** Link Budget Parameters [Kashitani, 2002]

Parameter	Value
Single User Downlink Data Rate	1.544 Mbps
Downlink Frequency	12.2 GHz
Required Bit Error Rate	$10^{-5}$
Modulation Scheme	QPSK
Required Signal-to-Noise Ratio	6.4 dB
Rain Loss + Link Margin	7.8 dB
Multi-Access Scheme	Spot Beams+FDMA, TDMA
Multi-Access Efficiency	90%
Minimum Required Elevation Angle	$10^\circ$
User Terminal Aperture Diameter	0.6 m
User Terminal Illumination Efficiency	60%
User Terminal System Noise Temperature	135 K

where  $\lambda$  is the downlink transmission frequency. Atmospheric loss  $L_A$  in the Ku-band may be estimated as a function of the elevation angle  $\epsilon$  between the user terminal and the satellite:

$$L_A = 10^{\frac{(-0.07)}{\sin \epsilon}} \quad (6.4)$$

Phased array antenna downlink  $P_D$  power is assumed to be two-thirds of the total payload power  $P_P$ :

$$P_D = 0.67P_P \quad (6.5)$$

The gain  $G_{PA}$  of the satellite phased array transmission antenna is:

$$G_{PA} = \eta_{PA} \frac{4\pi A_{PA}}{\lambda^2} \quad (6.6)$$

where  $\eta_{PA}$  is the phased array antenna efficiency and  $A_{PA}$  is the area of the phased array. Assuming a 60% user terminal illumination efficiency, the user terminal antenna gain  $G_u$

measures the amplification of the signal on the receiving end as a function of the user terminal antenna diameter  $D_u$ :

$$G_u = 0.6 \left( \frac{\pi D_u}{\lambda} \right)^2 \quad (6.7)$$

The total downlink capacity  $R_T$  of the satellite to the Earth grid can now be computed according to the link budget equation as:

$$R_T = \frac{P_D G_{PA} G_u L_S L_A L_{RM}}{SNR k_B T_u} \quad (6.8)$$

where  $L_{RM}$  is the rain loss plus link margin,  $SNR$  is the required signal-to-noise ratio,  $k_B$  is Boltzmann's constant, and  $T_u$  is the user terminal system noise temperature. Finally, the capacity (i.e. the number of T1 connections)  $C_{SAT}$  that can be placed on this link is:

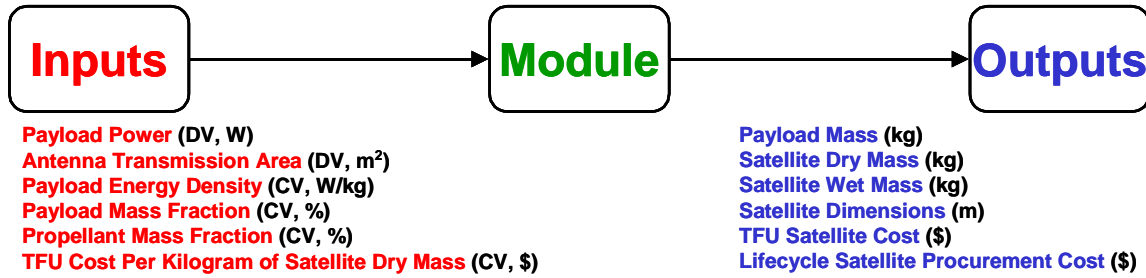
$$C_{SAT} = \frac{\eta_{ma} R_T}{R_u} \quad (6.9)$$

where  $\eta_{ma}$  is the multi-access scheme efficiency and  $R_u$  is the individual user data rate (i.e. T1 link = 1.544 Mbps). The calculations in Eqn. 6.3 through Eqn. 6.9 are repeated between each satellite and all of the user terminals in view of that satellite over every time step in the simulation to compute the maximum capacity of a broadband system architecture.

### **Payload & Bus Module**

Figure 6.10 lists key inputs and outputs of the payload and bus module. With knowledge of the phased array antenna characteristics and the assumed mass fractions, the satellite bus mass may be estimated. Key outputs include the payload mass, satellite dry mass, satellite wet mass, satellite dimensions, and theoretical first unit (TFU) satellite cost.

The payload mass  $M_p$  is estimated assuming an energy density of six watts per kilogram of phased array antenna mass:



**Figure 6.10** Payload & Bus Module Inputs and Outputs

$$M_P = \frac{P_P}{6} \quad (6.10)$$

Assuming a 33% payload mass fraction, the dry mass  $M_D$  of the satellite can be estimated as:

$$M_D = 3M_P \quad (6.11)$$

The wet mass  $M_W$  of the satellite is then estimated as:

$$M_W = \frac{M_D}{0.831} \quad (6.12)$$

This wet mass will be used downstream in the launch and operations module to compute the mass-limited capacity (i.e. number of satellites) for each of the launch vehicles under consideration. Based on a historical satellite density of 79 kg/m<sup>3</sup> [Larson & Wertz, 1992], an average value for satellites in orbits ranging from LEO to GEO, the volume  $V_S$  of the satellite may be estimated as:

$$V_S = \frac{M_W}{79} \quad (6.13)$$

The diameter  $D_S$  of the satellite bus is computed via another historical parametric relationship [Larson & Wertz, 1992]:

$$D_S = 0.25\sqrt[3]{M_W} \quad (6.14)$$

The height  $h_S$  of the satellite bus is calculated by dividing the volume of the bus by the cross-sectional area of the bus:

$$h_S = \frac{V_S}{\pi \left(\frac{D_S}{2}\right)^2} \quad (6.15)$$

The satellite volume and dimensions will be used downstream in the launch and operations module to compute the volume-limited payload fairing capacity of each of the launch vehicles under consideration.

Assuming a TFU cost density  $c_{TFU}$  of \$84,000/kg in constant year 2000 dollars, the TFU cost  $C_{S\_TFU}$  of a single broadband satellite is:

$$C_{S\_TFU} = c_{TFU} M_D \quad (6.16)$$

Once the TFU satellite cost is known, the total cost  $C_S$  for all of the satellites in the constellation may be calculated taking into account a learning curve [Larson & Wertz, 1992]. Accordingly,  $C_S$  may be computed as:

$$\begin{aligned} \beta_S &= 1 - \frac{\ln\left(\frac{100\%}{S_S}\right)}{\ln 2} \\ L_S &= n_S^{\beta_S} \\ C_S &= C_{S\_TFU} L_S \end{aligned} \quad (6.17)$$

where  $S_S$  is the learning curve slope in percent for the number of payloads being produced,  $L_S$  is the learning curve factor, and  $n_S$  is the total number of satellites required for the entire mission. In the constants vector,  $S_S$  is set to 95% when  $n_S$  is 10 or less, 90% when  $n_S$  is between 10 and 50, and 85% when  $n_S$  is greater than 50.

Finally, the non-recurring costs  $C_{S\_NR}$  of the satellites (i.e. research and development costs) estimated as:



$$C_{S\_NR} = f_{NR\_S} C_{S\_TFU} \quad (6.18)$$

where the non-recurring satellite cost factor  $f_{NR\_S}$  was set to a value of three in the constants vector.

### Launch and Operations Module

Figure 6.11 illustrates key inputs and outputs of the launch and operations module. With knowledge of the orbital characteristics of the constellation, mass and volume characteristics of the satellites, performance characteristics of the launch vehicles under consideration, ground stations costs, and labor costs; the lifecycle launch and operations costs may be estimated. Key outputs from this module include a launch vehicle selection strategy and the required number of groundstations and crew size to operate the system.

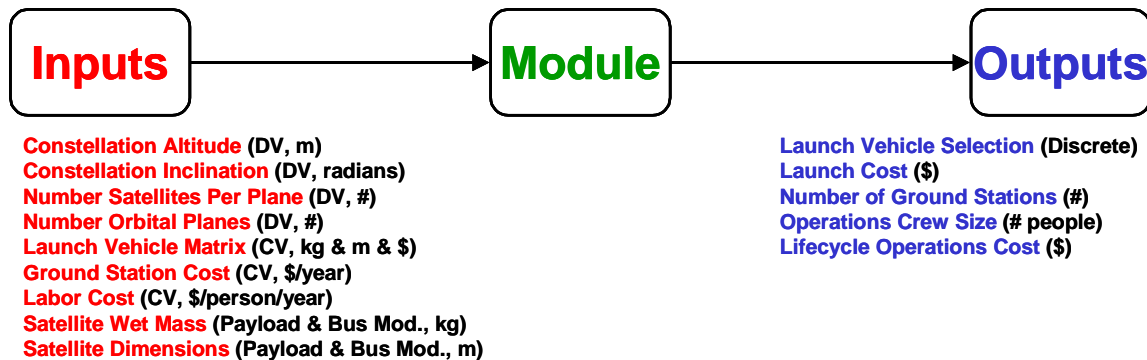


Figure 6.11 Launch and Operations Module Inputs and Outputs

The integer programming approach presented in Appendix B is used to solve the broadband launch vehicle selection problem as an optimization problem. The decision variables in the integer programming formulation of the launch vehicle selection problem are how many of each of the launch vehicles under consideration – the Ariane V, Atlas V, Delta IV, and Proton M – should be used to deploy the system. Table 6.5 lists the constants vector parameters of these four launch vehicles. The objective function is to select a launch vehicle(s) to deploy the broadband constellation at minimum cost. Finally, the operations

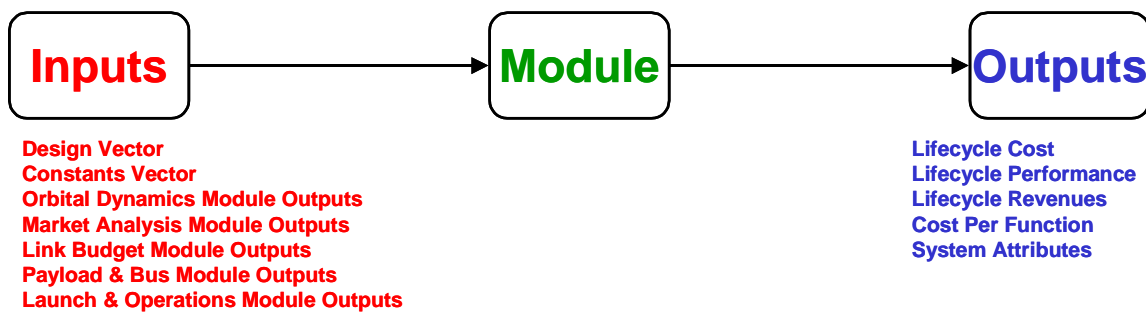
model estimates the number of groundstations and the crew size required to operate the system, and uses this information in turn to estimate the operations lifecycle cost.

**TABLE 6.5** Launch Vehicle Parameters [Kashitani, 2002]

Parameter	Ariane V	Atlas V	Delta IV	Proton M
Cost (\$M)	150	100	90	80
Payload Fairing Diameter (m)	5.4	5.0	5.0	4.4
Payload Fairing Height (m)	17.0	20.7	14.3	11.6
Launch Site Latitude (degrees)	5.2	28.5	28.5	28.5
Mass Delivered to 1000 km Orbit	17,856	13,060	8,060	5,174
Mass Delivered to 15,000 km Orbit	13,356	10,060	6660	4524
Mass Delivered to 35,000 km Orbit	7756	6060	4860	3624

### Systems Analysis Module

Figure 6.12 illustrates the inputs and outputs of the systems analysis module. With knowledge of the outputs from all the preceding modules in the broad GINA model, key system attributes that will be used to compare various conceptual design architectures may be computed. Key outputs include the lifecycle cost, lifecycle performance, lifecycle revenues, and the cost per function.

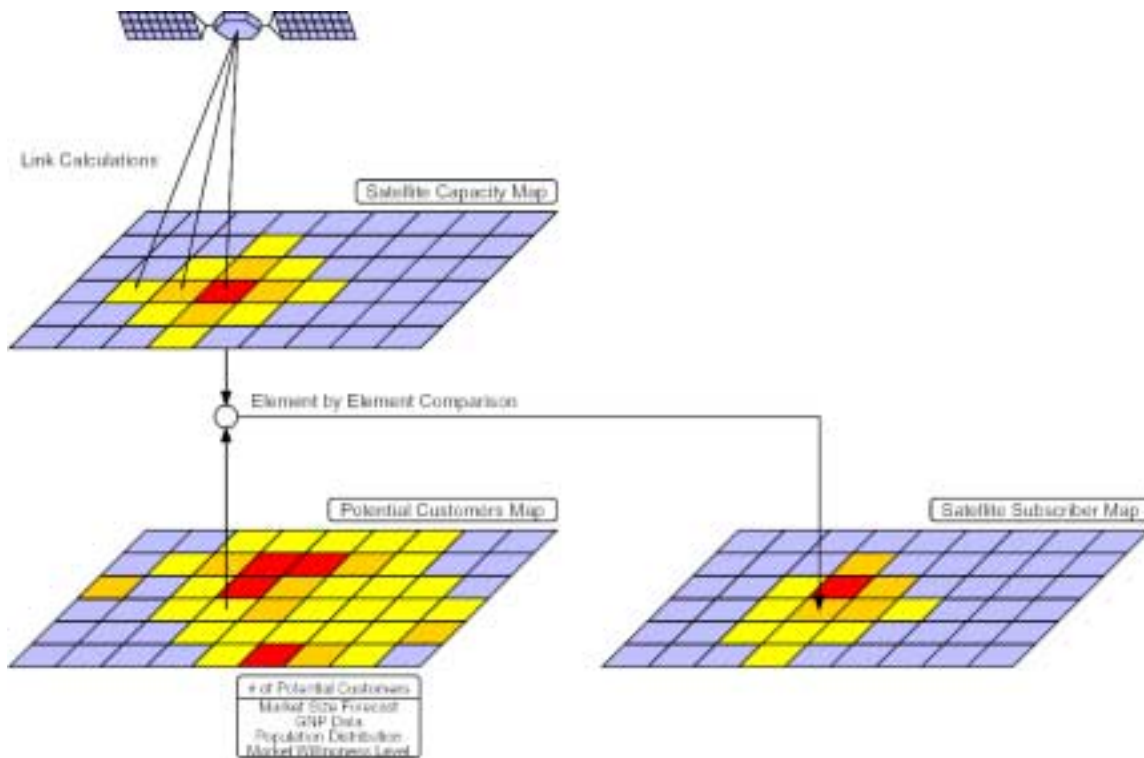


**Figure 6.12** Systems Analysis Module Inputs and Outputs

The total lifecycle cost  $\Phi_y$  of a broadband architecture over ten years is:

$$\sum_{y=1}^{10} \Phi_y = C_S + C_{S\_NR} + C_L + C_{OPS} \quad (6.19)$$

where  $y$  is the year in the mission,  $C_S$  is the cumulative cost for all of the satellites in the constellation,  $C_{S\_NR}$  is the total non-recurring cost,  $C_L$  is the launch cost as computed by the integer programming model in Appendix B, and  $C_{OPS}$  is the lifecycle operations cost as computed in the launch & operations modules. As illustrated in Figure 6.13, the lifecycle performance of a broadband satellite architecture is computed by matching the network capacity to the market demand to determine the total number of paying subscribers for the system. For the details behind this capacity-demand matching algorithm and the computation of the satellite subscriber map, please see the reference by Kashitani [Kashitani, 2002].



**Figure 6.13** System Performance: Matching Network Capacity to Market Demand [Kashitani, 2002]

The CPF metric, which for this mission is the cost per subscription (i.e. billable T1-minute) required for the project to achieve a zero net present value over the ten year mission duration, is computed assuming a 30% expected internal rate of return  $i_{RR}$  and a 1.7% inflation rate  $i_{Inf}$  as set in the constants vector. In this case, a zero net present value yields a 30% return on the investment made to design, construct, deploy, and operate the broadband satellite network. The systems analysis module assumes that the charge for the network service in year 2000 dollars  $C_{S2000}$  is constant during the ten-year operations period. This means that the subscription price in nominal year dollars  $C_{SN}$  increases with inflation:

$$C_{SN}(y) = C_{S2000}(1 + i_{Inf})^y \quad (6.20)$$

where  $y$  is the year of the mission. The present value of the subscription price  $C_{SPV}$  is calculated by discounting the nominal charge each year by the internal rate of return that year:

$$C_{SPV}(y) = \frac{C_{SN}}{(1 + i_{RR})^y} \quad (6.21)$$

Combining Eqn. 6.20 with Eqn. 6.21, the present value of the subscription price may be expressed as:

$$C_{SPV} = C_{S2000} \left( \frac{1 + i_{Inf}}{1 + i_{RR}} \right)^y \quad (6.22)$$

Assuming the system revenue comes exclusively from subscription fees, then the present value of the total revenue  $R_{PV}$  is the cumulative sum of the product of the present value subscription price each year and the number of subscribers  $N_S$  each year:

$$R_{PV} = \sum_{y=1}^{10} C_{SPV}(y)N_S(y) \quad (6.23)$$

Substituting Eqn. 6.22 into Eqn. 6.23 yields:

$$R_{PV} = C_{S2000} \left[ \sum_{y=1}^{10} \left( \frac{1 + i_{Inf}}{1 + i_{RR}} \right)^y N_S(y) \right] \quad (6.24)$$

Because the commercial mission must achieve a zero net present value after a ten year operational life, the present value total revenue is equal to the present value of the total system cost  $C_{TPV}$  (i.e. the present value of the expression in Eqn. 6.19). Substituting  $C_{TPV}$  for  $R_{PV}$  and solving Eqn. 6.24 for  $C_{S2000}$  yields the cost per subscription in year 2000 constant dollars:

$$C_{S2000} = \frac{C_{TPV}}{\sum_{y=1}^{10} \left( \frac{1 + i_{Inf}}{1 + i_{RR}} \right)^y N_S(y)} \quad (6.25)$$

### **Benchmarking the Broadband GINA Model**

The broadband GINA model has been benchmarked against point designs proposed by Boeing [Boeing, 1999], Hughes Communications [Hughes Communications, 1999], and a European consortium [SkyBridge, 1997; SkyBridge, 1999]. The results of evaluating each of these design using the GINA model were compared to the broadband system parameters generated by the companies. When discrepancies were found, they were addressed by first examining the assumptions behind the model and then modifying the required module(s) to increase model fidelity. The goal of this verification process was to gain confidence in the results produced by the broadband GINA model, not to make the output parameters exactly match those from the other designs. Overall, the validation test cases verified the fidelity of broadband GINA model. Please see the reference by Kashitani [Kashitani, 2002] for a complete description of the benchmarking exercise.

### **6.2.6 GINA Step 6 – Explore the System Trade Space**

With the GINA model now complete, the broadband trade space may be explored. This will be done in step two of the MMDOSA methodology.

### 6.3 MMDOSA Step 2 – Perform Univariate Studies

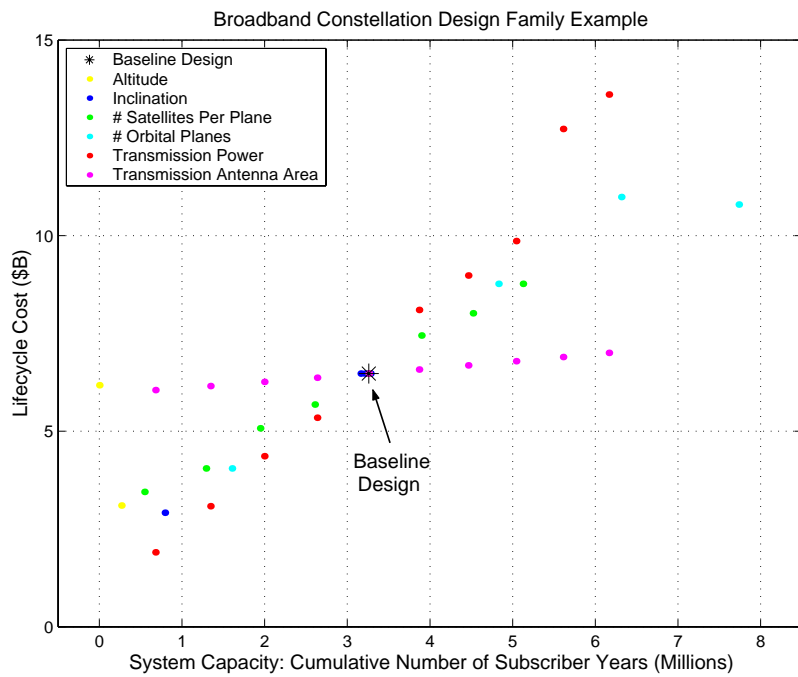
Univariate studies may be used to find the general shape of a "design family" within the system trade space. A design family is created by establishing a baseline design and then varying each parameter within the design vector across its entire range while holding all other components of the design vector at their baseline values. Table 6.6 lists the baseline design vector for the design family illustrated in Figure 6.14.

**TABLE 6.6** Baseline Broadband Design Vector

Design Vector	Value
Constellation Altitude	MEO
Constellation Inclination	45°
# Satellites Per Plane	5
# Orbital Planes	4
Payload Power	5 kW
Antenna Transmission Area	2.5 m <sup>2</sup>

Figure 6.14 illustrates a sample broadband design family, with the system capacity on the x-axis and lifecycle cost on the y-axis. Each dot represents a different broadband system architecture within the same design family. The black asterisk represents the baseline design (Table 6.6) for this particular family. Of all the design vector variables, the area of the phased array antenna has the least effect on system lifecycle cost while still having a dramatic effect on system capacity. The phased array antenna payload power exerts a similar effect on system capacity, but with a much greater difference in lifecycle cost. Varying the number of satellites per orbital plane and the number of orbital planes (i.e. varying the total number of satellites in the constellation) has a nearly equal linear effect on both capacity and cost until the capacity of the system exceeds the size of the serviceable market, after which adding more satellites increases the lifecycle cost with no further improvement in performance as measured by the number of subscriptions supported. Finally, the orbital altitude and inclination of the constellation exert a nonlinear effect on both cost and performance, with equivalent sized constellations in MEO with medium

inclinations providing the highest capacity. Thus, it appears that, at least in this local trade space, that the phased array antenna payload power exerts the greatest influence on both system performance and cost. This issue will be revisited within the context of the global trade space in Section 6.6. Figure 6.15 through Figure 6.20 illustrate univariate studies executed on the broadband GINA model to verify model fidelity and gain an initial feel for the trade space.



**Figure 6.14** Example Broadband Design Family

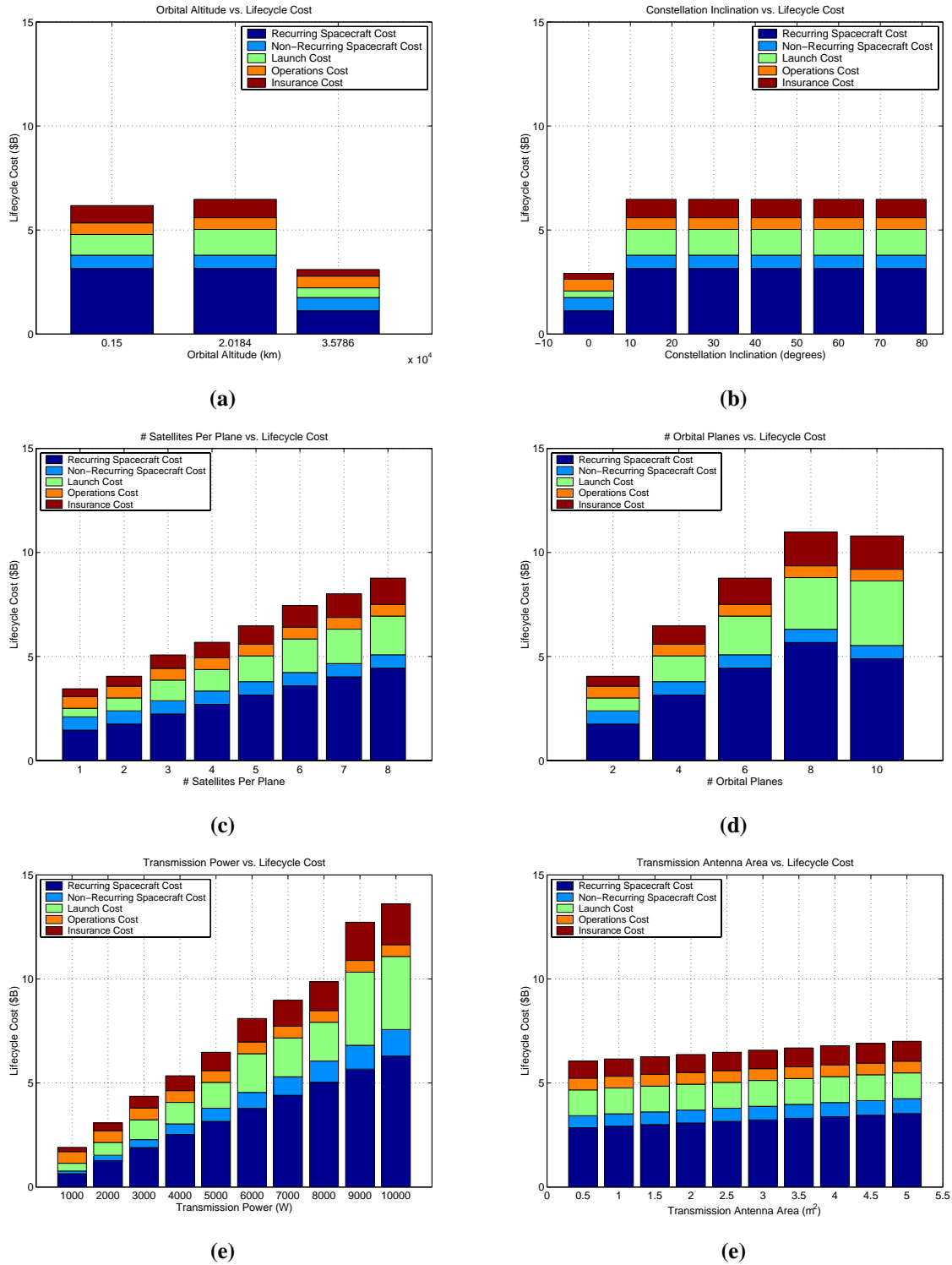
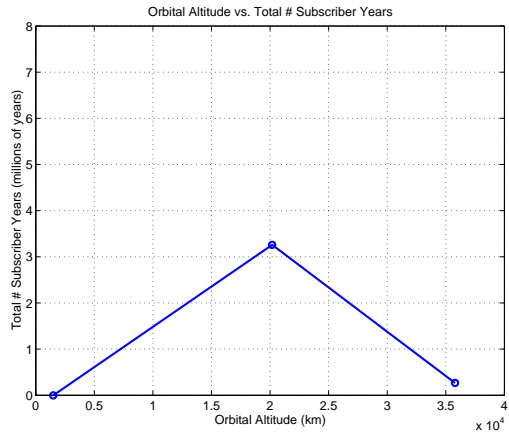
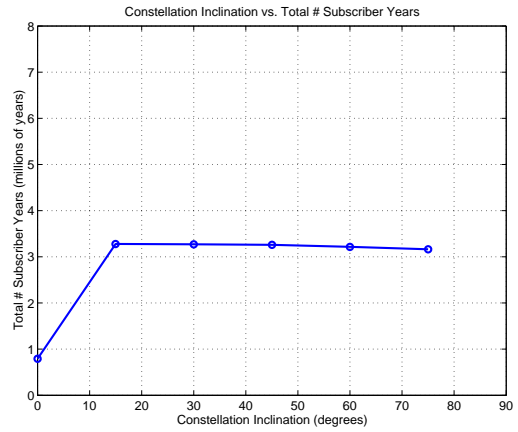


Figure 6.15 Lifecycle Cost Univariate Studies

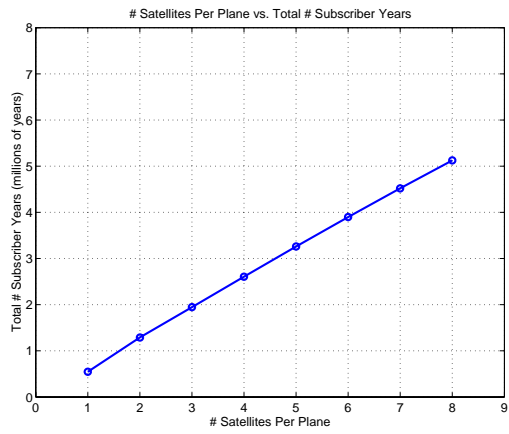




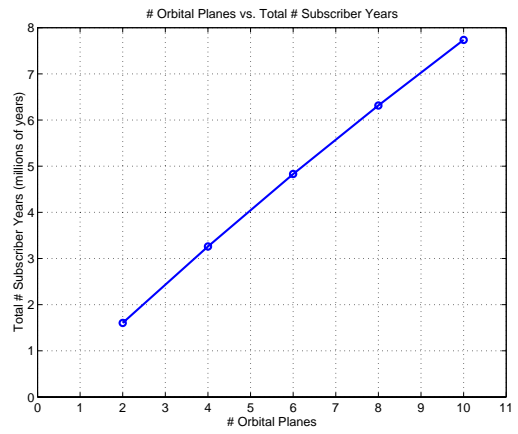
(a)



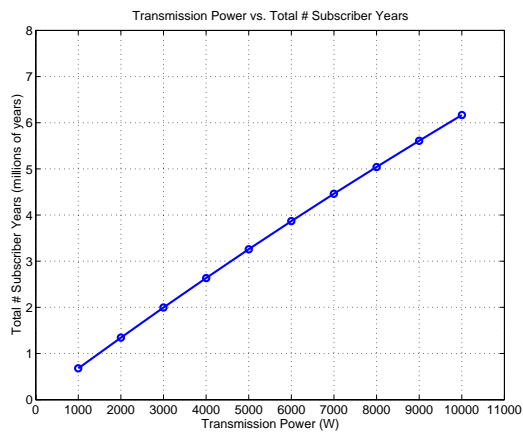
(b)



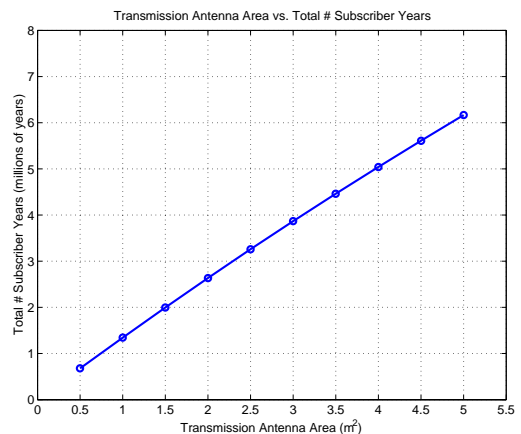
(c)



(d)

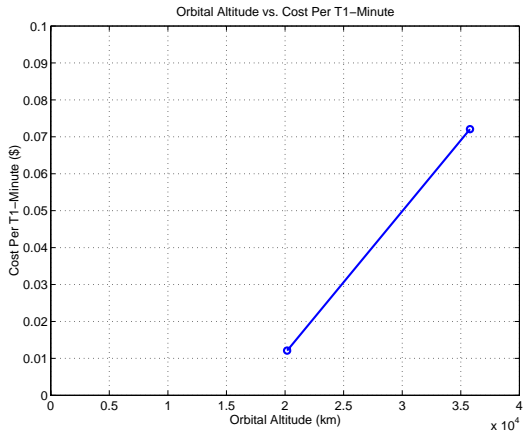


(e)

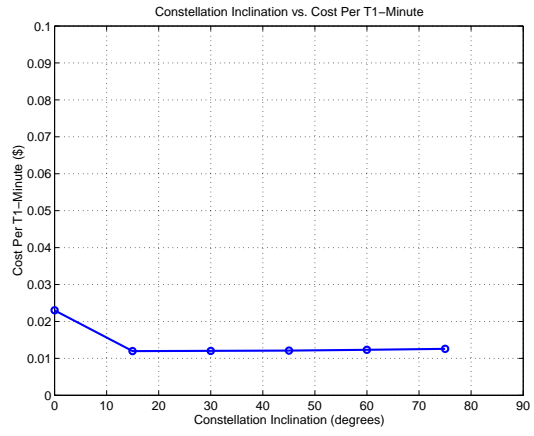


(e)

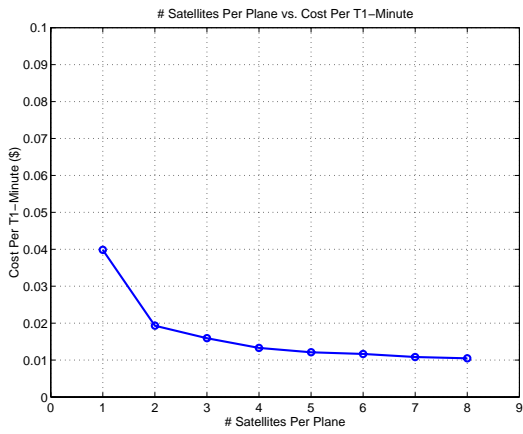
**Figure 6.16** Total Number of Subscriber Years (i.e. System Performance) Univariate Studies



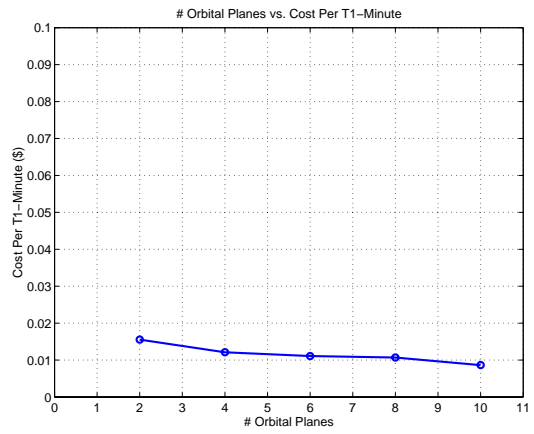
(a)



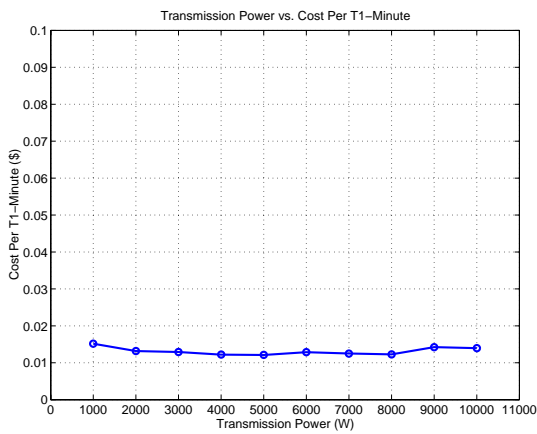
(b)



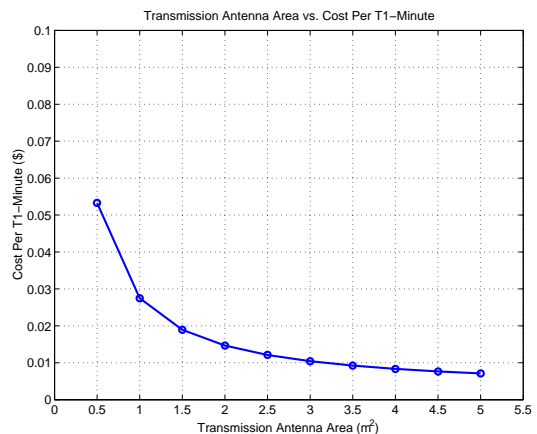
(c)



(d)

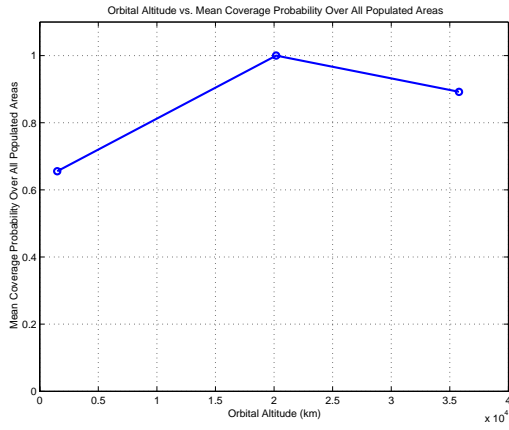


(e)

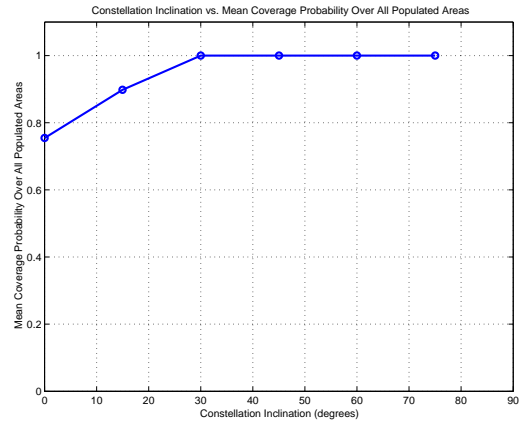


(e)

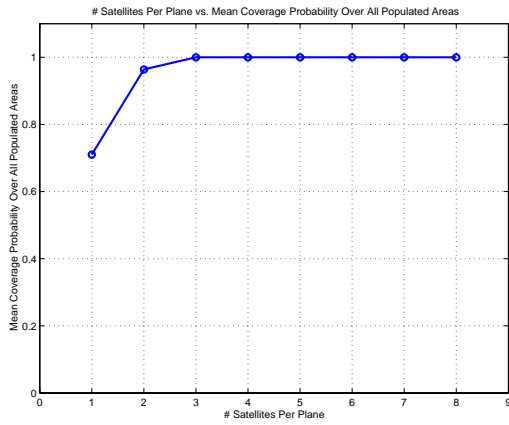
Figure 6.17 Cost Per T1-Minute Univariate Studies



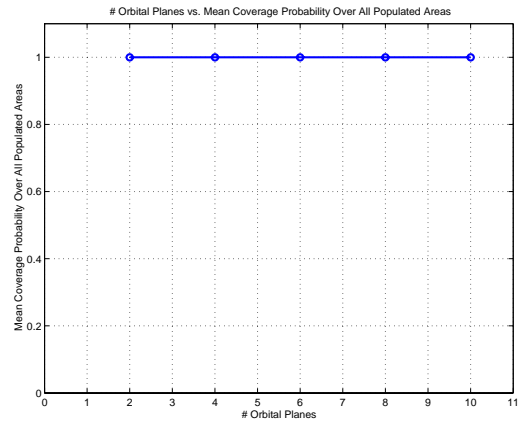
(a)



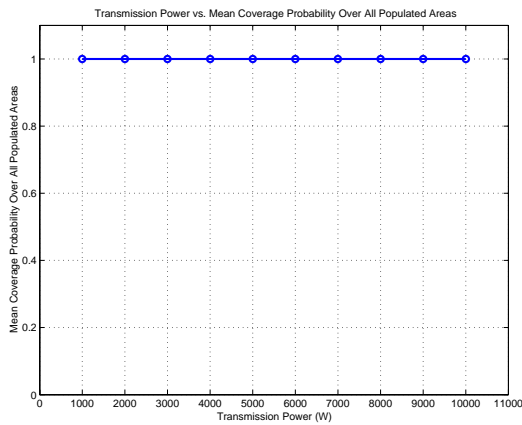
(b)



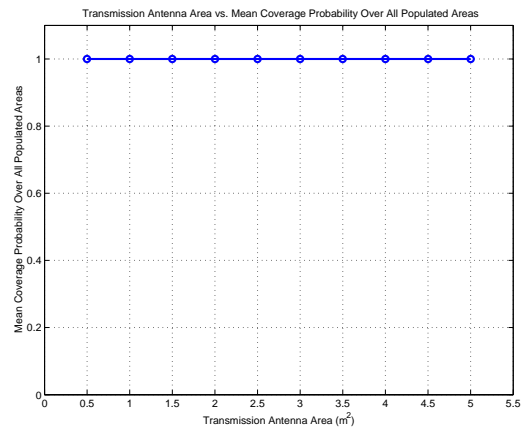
(c)



(d)

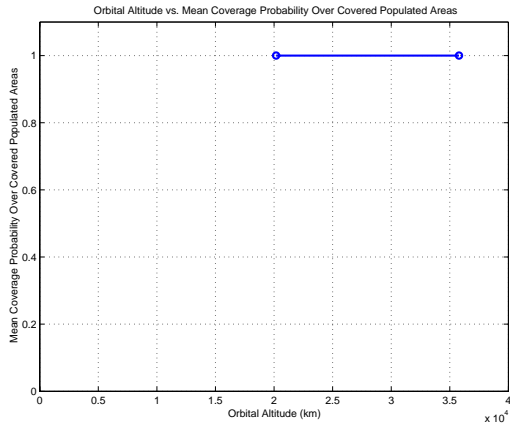


(e)

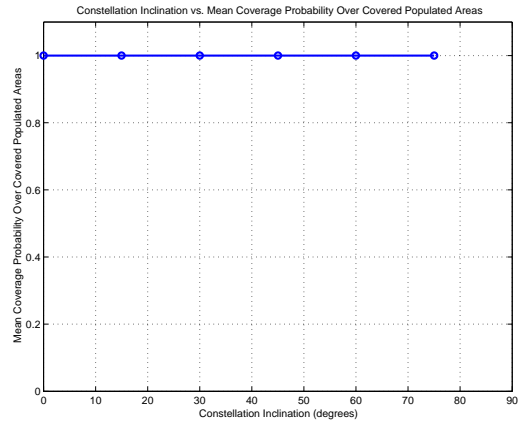


(e)

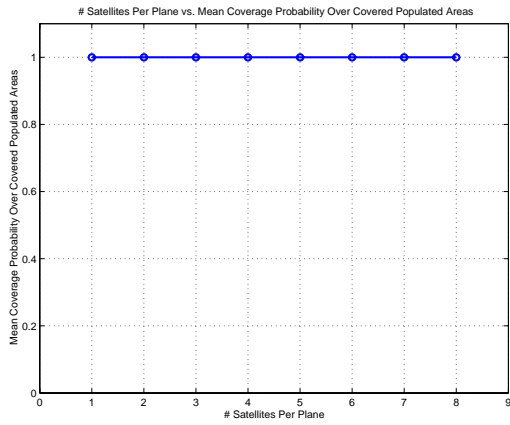
**Figure 6.18** Mean Coverage Probability Over All Populated Areas (i.e. Entire Potential Market) Univariate Studies



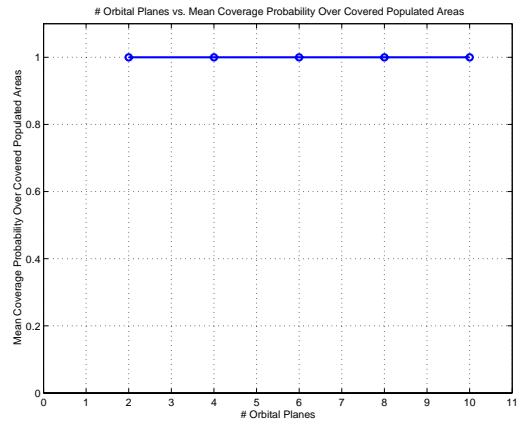
(a)



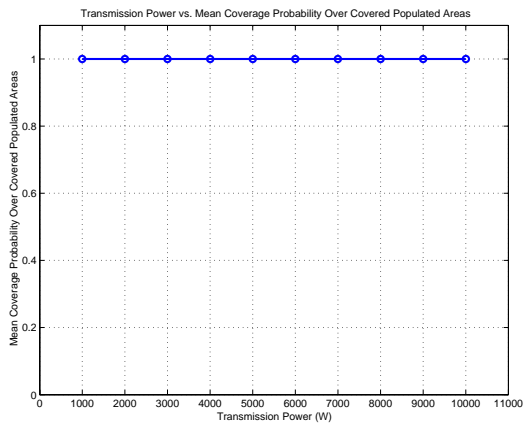
(b)



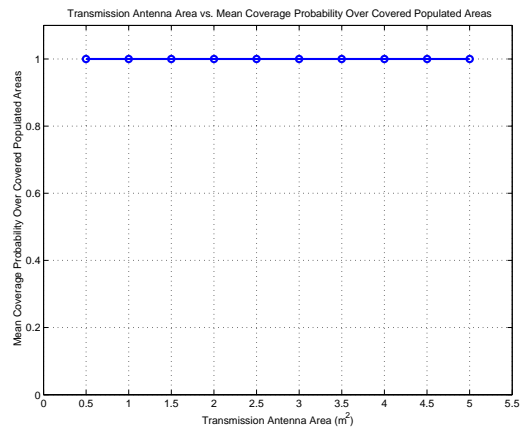
(c)



(d)

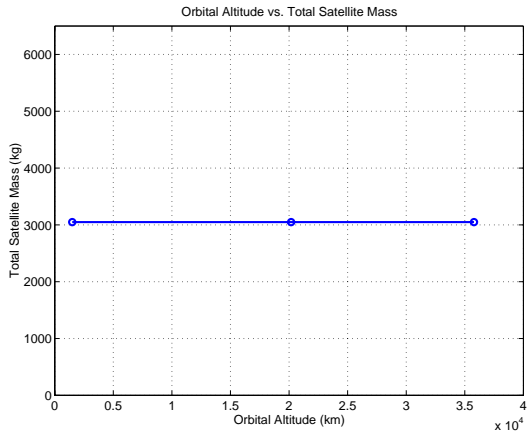


(e)

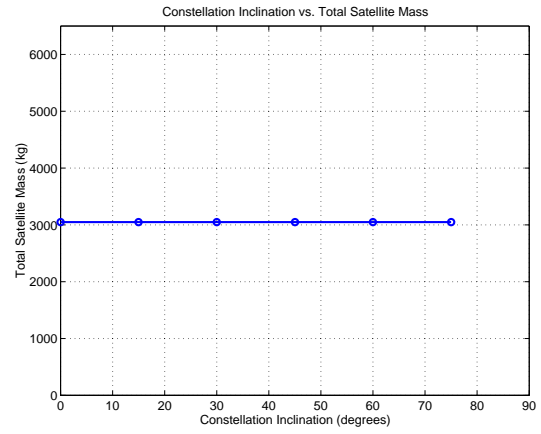


(e)

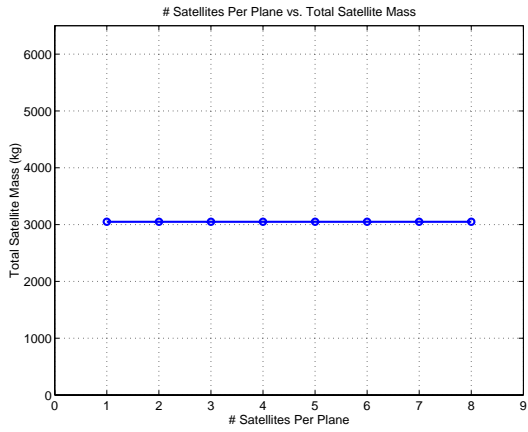
Figure 6.19 Mean Coverage Probability Over Served Populated Areas (i.e. Availability) Univariate Studies



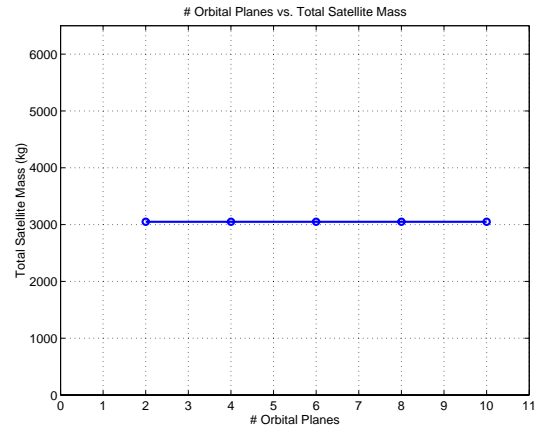
(a)



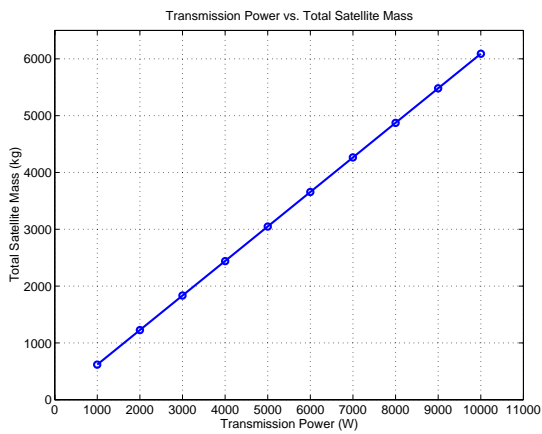
(b)



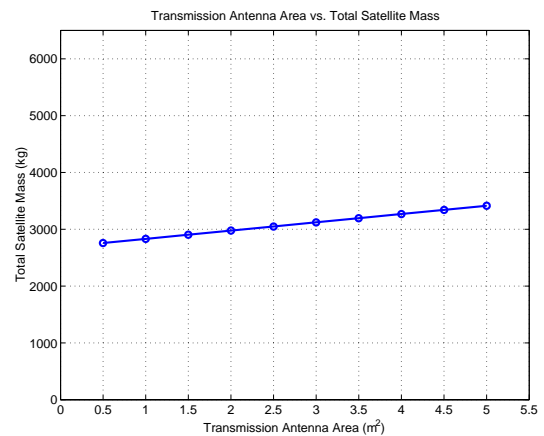
(c)



(d)



(e)



(e)

Figure 6.20 Satellite Mass Univariate Studies

## 6.4 MMDOSA Step 3 – Take a Random Sample of the Trade Space

Within the broadband global trade space of 42,400 architectures, 200 architectures – 0.47% of the trade space – were randomly sampled. Of these 200 architectures, 152 were feasible conceptual design solutions based on the 98% availability over served customers requirement. Figure 6.21 illustrates the results of the random sample for six architecture attributes. Table 6.7 summarizes the statistics from this random sample for each metric which the broadband design is simultaneously optimized with respect to in Section 6.5. Please see Section 6.5 for details on how these statistics from the random sample are used to tailor the MDO search algorithms.

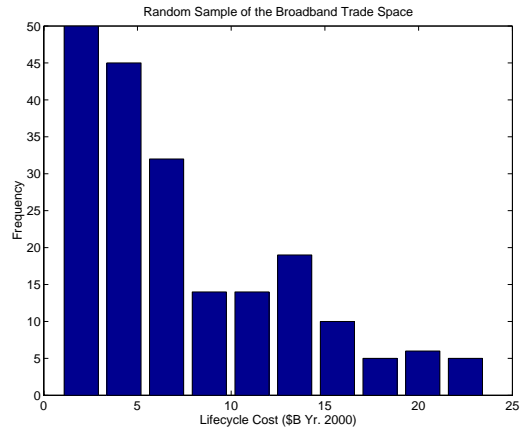
**TABLE 6.7** Statistics from a Random Sample of the Broadband Trade Space

Parameter	Lifecycle Cost (\$B)	Performance (Subscriber Years x10 <sup>6</sup> )	Cost Per T1-Minute (\$)	Global Availability (%)	Availability for Served Customers (%)
Maximum	23.6	29.5	2.28	100	100
Minimum	0.9	0.0	0.0013	5.0	98.3
Range	22.7	29.5	2.27	95.0	1.7
Mean	7.6	4.8	0.076	71.7	99.9
Median	6.1	0.8	0.031	79.2	100
Standard Deviation	5.6	8.5	0.20	28.2	0.2

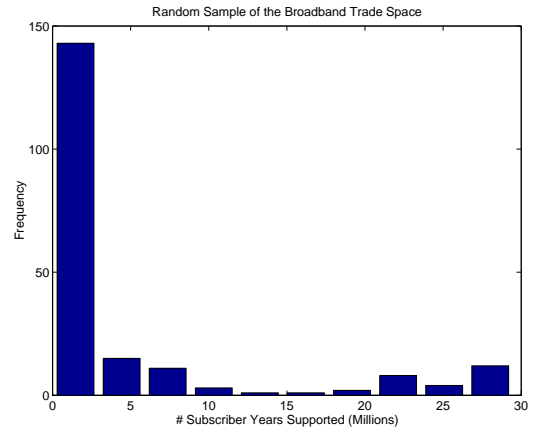
These statistics can be used to compute the 95% confidence interval (95%CI) for each parameter  $i$  in the objective function:

$$95\% \text{ CI} = \bar{x}_i \pm \frac{2v(x_i)}{\sqrt{n}} \quad (6.26)$$

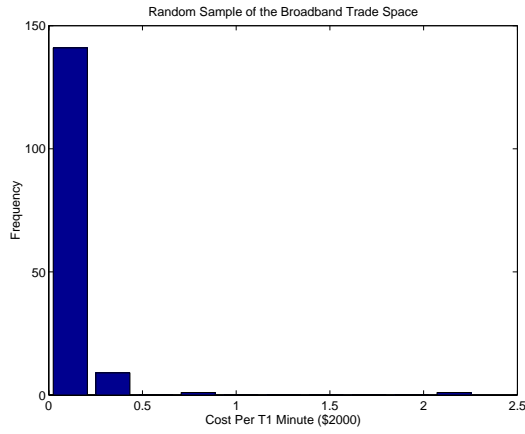
where  $\bar{x}_i$  is the sample mean value for parameter  $i$ ,  $v(x_i)$  is the sample standard deviation for parameter  $i$ , and  $n$  is the size of the random sample. Table 6.8 lists the 95%CI of the mean value for the different architecture parameters that were measured in the random sample. This information tells the systems engineer what the likely characteristics of an



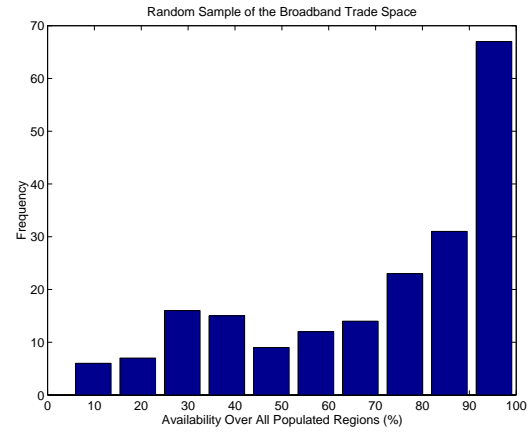
(a)



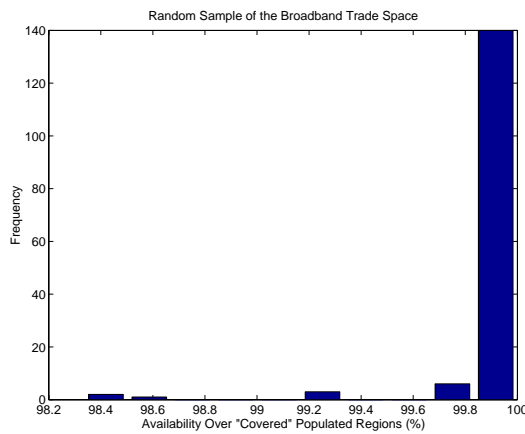
(b)



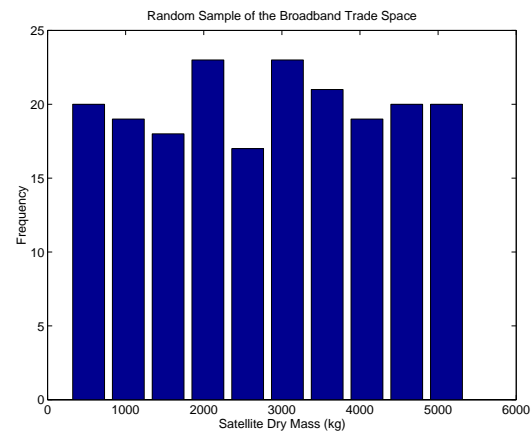
(c)



(d)



(e)



(e)

**Figure 6.21** Random Sample Results for Lifecycle Cost (a), System Performance (b), Cost Per T1-Minute (c), Global Availability (d), Served Customers Availability (e), and Satellite Mass (e)

"average" design will be. For the broadband satellite communications mission, it can be said with 95% confidence that a randomly selected set of system architectures will exhibit on average a lifecycle cost of approximately \$7.6B, a performance of 4.8 million cumulative subscriber years, a cost per T1-minute of \$0.08, a global population coverage probability of 72%, and a customer availability of 99.9%.

**TABLE 6.8** Broadband 95% CI's for the Global Trade Space Mean

Parameter	95%CI
Lifecycle Cost	\$6.83B - \$8.41B
Performance	3.63 - 6.03 x10 <sup>6</sup> subscriber years
Cost Per T1-Minute	\$0.05 - \$0.11
Global Availability.	68% - 76%
Availability for Served Customers	99.8% - 100%

The systems engineer is not interested in creating an "average" design for the broadband system, however. Rather, the systems engineer wishes to find the best system architecture(s) based on the metric(s) of interest to the customer. Thus, the broadband architectures found via the MDO algorithms in step four of the MMDOSA methodology should rate much better than the 95%CI parameter values listed in Table 6.8. In fact, the data gathered in the random sample of the trade space (Table 6.7) can be used to place initial bounds on each of the design parameters to be optimized. For example, the minimum CPF broadband architectures found via MDO algorithms should yield a lower CPF than the minimum CPF value of \$0.0013 per T1-minute in Table 6.7, which was the lowest CPF value measured in the random sample.

## 6.5 MMDOSA Step 4 – Formulate and Apply MDO Algorithm(s)

Three separate MDO approaches have been applied to the exploration of the broadband satellite communications trade space. These approaches are summarized in the test matrix in Table 6.9. In the first test case, a single objective optimization approach is used to



search the broadband global trade space for architectures that minimize the cost per function metric. For the broadband mission, the CPF metric is defined as the cost per T1-minute that needs to be charged to achieve a zero net present value for the project after ten years of operations; assuming satellite-based systems will penetrate up to 1% of the global broadband market, inflation will remain steady at 1.7%, and that the investors in the broadband venture desire a 30% internal rate of return on their initial investment. In the second test case, the design scope is expanded from that of a single objective optimization problem to a more difficult multiobjective optimization problem in which the systems engineer simultaneously attempts to minimize lifecycle cost and maximize system performance. For the broadband mission, system performance is defined as the cumulative number of billable T1-minutes the architecture provides over its ten-year life. In the third case study, the number of decision criteria in the multiobjective optimization problem increases from two to four and includes the desire to maximize system availability over the market served and also maximize system availability over the entire populated globe.

### **6.5.1 Approach 1 – Single Objective Optimization**

#### **Objective**

In the first stage of the test matrix, the objective is to find, without completely enumerating the trade space, the broadband architectures that minimize the CPF metric – the cost per billable T1-minute required to achieve zero net present value. This is done by applying the single objective simulated annealing algorithm described in Section 3.5.1. The sole objective is to minimize the CPF metric under the following five business case scenarios:

1. Commercial system with satellite-only service.
2. Commercial system with satellite-only service promising global population coverage.
3. Commercial satellite system with a joint terrestrial network service provider partner.
4. Commercial satellite system promising global population coverage with a joint terrestrial network service provider partner.

**TABLE 6.9** MDO Test Matrix for the Broadband Case Study

<b>Test Case</b>	<b>Structure</b>
<b>Approach 1</b>	
Type	Single Objective Optimization
Objective Function	Minimize CPF (i.e. Cost Per Billable T1-Minute)
Trade Space Description	Broadband Global Trade Space
Trade Space Size	42,400 system architectures
<b>Approach 2</b>	
Type	Multiobjective Optimization
Objective Function	Minimize Lifecycle Cost and Maximize System Performance
Trade Space Description	Broadband Global Trade Space
Trade Space Size	42,400 system architectures
<b>Approach 3</b>	
Type	Multiobjective Optimization
Objective Function	Minimize Lifecycle Cost and Maximize System Performance and Maximize Market Availability and Maximize Global Availability
Trade Space Description	Broadband Global Trade Space
Trade Space Size	42,400 system architectures

5. Commercial system seeking to capture a U.S. military contract.

These five scenarios illustrate the ability of the MMDOSA methodology to search the trade space for different customers and business plans under alternative market scenarios by simply changing the formulation of the optimization problem – in this case the constraint equations. This first design problem may be characterized as a single objective DSS architecture conceptual design problem.

### **Problem Formulation**

The first business case scenario involves a commercial system with satellite-only service. No requirements are placed on global coverage, and the MDO algorithm is free to determine whether a regional coverage system or a global coverage system makes more sense from a business case perspective. The scenario one broadband architecture design single objective optimization problem may be formulated mathematically as:

$$\begin{aligned}
 \text{Objective:} & \quad \text{Min} \left( \frac{\sum_{y=1}^{10} (\Phi_y(\Gamma))(1+r)^{-y}}{\sum_{y=1}^{10} \left( \frac{1+i_{Inf}}{1+i_{RR}} \right)^y N_S(y)} \right) \\
 \text{Constraints:} & \quad \text{Subject to} \\
 \text{Isolation} & \quad \text{MAE} \geq 90\% \\
 & \quad \text{SNR} \geq 6.4 \text{ dB} \\
 & \quad L_M \geq 7.8 \text{ dB} \\
 \text{Integrity} & \quad \text{BER} \leq 10^{-5} \\
 \text{Rate} & \quad R \geq 1.54 \text{ Mbps per link} \\
 \text{Availability} & \quad \epsilon_{\min} \geq 10^\circ \\
 & \quad A_{vm} \geq 98\% \\
 \text{Bounds} & \quad 0^\circ \leq \gamma_2 \leq 75^\circ \\
 & \quad 1 \leq \gamma_3 \leq 8 \\
 & \quad 2 \leq \gamma_4 \leq 8 \\
 & \quad 1 \text{ kW} \leq \gamma_5 \leq 10 \text{ kW} \\
 & \quad 0.5 \text{ m}^2 \leq \gamma_6 \leq 5.0 \text{ m}^2
 \end{aligned} \tag{6.27}$$

where  $y$  is the year in the mission,  $\Phi_y(\Gamma)$  is the cost in year  $y$  of architecture  $\Gamma$ ,  $r$  is the discount rate,  $i_{Inf}$  is the inflation rate,  $i_{RR}$  is the internal rate of return, and  $N_S(y)$  is the number of subscriptions the architecture can support in year  $y$ , MAE is the multi-access efficiency, SNR is the required T1 link signal-to-noise ratio,  $L_M$  is the link margin to account for rain loss and other losses, BER is the bit error rate,  $R$  is the data transfer rate,  $\epsilon_{\min}$  is the minimum viewing elevation angle to establish a link between a user terminal and satellite,  $A_{vm}$  is the system availability over the market served, and  $\gamma$ 's are the different elements of the design vector as defined in Table 6.2.

The second business case scenario also involves a commercial system with satellite-only service, but promises global population coverage. This new twist in the business plan mandates a constellation geometry that provides global coverage, eliminating the less

costly regional systems from consideration. The scenario two broadband architecture design single objective optimization problem may be formulated mathematically as:

$$\begin{aligned}
 \text{Objective:} & \quad \text{Min} \left( \frac{\sum_{y=1}^{10} (\Phi_y(\Gamma))(1+r)^{-y}}{\sum_{y=1}^{10} \left( \frac{1+i_{Inf}}{1+i_{RR}} \right)^y N_S(y)} \right) \\
 \text{Constraints:} & \quad \text{Subject to} \\
 \text{Isolation} & \quad \text{MAE} \geq 90\% \\
 & \quad \text{SNR} \geq 6.4 \text{ dB} \\
 & \quad L_M \geq 7.8 \text{ dB} \\
 \text{Integrity} & \quad \text{BER} \leq 10^{-5} \\
 \text{Rate} & \quad R \geq 1.54 \text{ Mbps per link} \\
 \text{Availability} & \quad \epsilon_{\min} \geq 10^\circ \\
 & \quad A_{vm} \geq 98\% \\
 & \quad A_{vgp} \geq 100\% \\
 \text{Bounds} & \quad 0^\circ \leq \gamma_2 \leq 75^\circ \\
 & \quad 1 \leq \gamma_3 \leq 8 \\
 & \quad 2 \leq \gamma_4 \leq 8 \\
 & \quad 1 \text{ kW} \leq \gamma_5 \leq 10 \text{ kW} \\
 & \quad 0.5 \text{ m}^2 \leq \gamma_6 \leq 5.0 \text{ m}^2
 \end{aligned} \tag{6.28}$$

where  $A_{vgp}$  is the availability of the broadband service over all the populated regions of the Earth. The addition of this new availability constraint into the problem formulation mandates global population coverage be achieved for an architecture to be considered a feasible solution to the conceptual design problem.

The third business case scenario involves a commercial satellite system that is partnered with a terrestrial network service provider. While it is preferable to deliver the broadband service via satellite, the information may be routed via terrestrial networks during instances when none of the satellites in the constellation are available over a given user

terminal. This results in the relaxation of the system availability over the market served constraint, which should in turn make less costly architectures feasible – but not necessarily business savvy – solutions to the conceptual design problem. As in business case scenario one, the MDO algorithm is free to determine whether a regional coverage system versus a global coverage system makes more sense from a business case perspective. The scenario three broadband architecture design single objective optimization problem may be formulated mathematically as:

$$\begin{aligned}
 \text{Objective:} & \quad \text{Min} \left( \frac{\sum_{y=1}^{10} (\Phi_y(\Gamma))(1+r)^{-y}}{\sum_{y=1}^{10} \left(\frac{1+i_{Inf}}{1+i_{RR}}\right)^y N_S(y)} \right) \\
 \text{Constraints:} & \quad \text{Subject to} \\
 \text{Isolation} & \quad \text{MAE} \geq 90\% \\
 & \quad \text{SNR} \geq 6.4 \text{ dB} \\
 & \quad L_M \geq 7.8 \text{ dB} \\
 \text{Integrity} & \quad \text{BER} \leq 10^{-5} \\
 \text{Rate} & \quad R \geq 1.54 \text{ Mbps per link} \\
 \text{Availability} & \quad \epsilon_{\min} \geq 10^\circ \\
 & \quad A_{vm} \geq 75\% \\
 \text{Bounds} & \quad 0^\circ \leq \gamma_2 \leq 75^\circ \\
 & \quad 1 \leq \gamma_3 \leq 8 \\
 & \quad 2 \leq \gamma_4 \leq 8 \\
 & \quad 1 \text{ kW} \leq \gamma_5 \leq 10 \text{ kW} \\
 & \quad 0.5 \text{ m}^2 \leq \gamma_6 \leq 5.0 \text{ m}^2
 \end{aligned} \tag{6.29}$$

Notice the reduction in the constraint on the system availability over the market served  $A_{vm}$  from 98% in business case scenario one to 75%.

The fourth business case scenario also involves a commercial satellite system that is partnered with a terrestrial network service provider, but promises global population coverage.

This new requirement in turn mandates a constellation geometry that provides global coverage, eliminating the less costly regional systems from consideration. The scenario four broadband architecture design single objective optimization problem may be formulated mathematically as:

$$\begin{array}{ll}
 \text{Objective:} & \text{Min} \left( \frac{\sum_{y=1}^{10} (\Phi_y(\Gamma))(1+r)^{-y}}{\sum_{y=1}^{10} \left( \frac{1+i_{Inf}}{1+i_{RR}} \right)^y N_S(y)} \right) \\
 \text{Constraints:} & \text{Subject to} \\
 \text{Isolation} & \text{MAE} \geq 90\% \\
 & \text{SNR} \geq 6.4 \text{ dB} \\
 & L_M \geq 7.8 \text{ dB} \\
 \text{Integrity} & \text{BER} \leq 10^{-5} \\
 \text{Rate} & R \geq 1.54 \text{ Mbps per link} \\
 \text{Availability} & \epsilon_{\min} \geq 10^\circ \\
 & A_{vm} \geq 75\% \\
 & A_{vgp} \geq 100\% \\
 \text{Bounds} & 0^\circ \leq \gamma_2 \leq 75^\circ \\
 & 1 \leq \gamma_3 \leq 8 \\
 & 2 \leq \gamma_4 \leq 8 \\
 & 1 \text{ kW} \leq \gamma_5 \leq 10 \text{ kW} \\
 & 0.5 \text{ m}^2 \leq \gamma_6 \leq 5.0 \text{ m}^2
 \end{array} \tag{6.30}$$

Notice the reduction in the constraint on the system availability over the market served  $A_{vm}$  from 98% in business case scenario one to 75% and the addition of the new availability  $A_{vgp}$  constraint into the problem formulation to mandate global population coverage be achieved in order for an architecture to be considered a feasible solution to the conceptual design problem.

Finally, the fifth business case scenario involves a commercial satellite system seeking to capture a U.S. military contract. To meet the military’s stringent requirements for battle-field operations, the system must provide a greater link margin to ensure the signal cannot be jammed. Further, the system must provide 100% availability over 100% of the Earth’s surface area to provide the capability to support a theater of operations anywhere in the world. The scenario five broadband architecture design single objective optimization problem may be formulated mathematically as:

$$\begin{array}{ll}
 \text{Objective:} & \text{Min} \left( \frac{\sum_{y=1}^{10} (\Phi_y(\Gamma))(1+r)^{-y}}{\sum_{y=1}^{10} \left(\frac{1+i_{Inf}}{1+i_{RR}}\right)^y N_S(y)} \right) \\
 \text{Constraints:} & \text{Subject to} \\
 \text{Isolation} & \text{MAE} \geq 90\% \\
 & \text{SNR} \geq 6.4 \text{ dB} \\
 & L_M \geq 18 \text{ dB} \\
 \text{Integrity} & \text{BER} \leq 10^{-5} \\
 \text{Rate} & R \geq 1.54 \text{ Mbps per link} \\
 \text{Availability} & \epsilon_{\min} \geq 10^\circ \\
 & A_{vm} \geq 100\% \\
 & A_{vgl} \geq 100\% \\
 \text{Bounds} & 0^\circ \leq \gamma_2 \leq 75^\circ \\
 & 1 \leq \gamma_3 \leq 8 \\
 & 2 \leq \gamma_4 \leq 8 \\
 & 1 \text{ kW} \leq \gamma_5 \leq 10 \text{ kW} \\
 & 0.5 \text{ m}^2 \leq \gamma_6 \leq 5.0 \text{ m}^2
 \end{array} \tag{6.31}$$

where  $A_{vgl}$  is the availability of the broadband service over the entire surface area of the Earth. Notice the dramatic increase in the link margin requirement from 7.8 dB to 18 dB and the 100% market served and global coverage requirements. To evaluate a broadband

system architecture on the basis of the objective function and constraints in Eqn. 6.27 through Eqn. 6.31, the design vector is entered into the broadband GINA model. Table 6.2 lists the design vector variables along with the allowable range for each variable's value.

### Algorithm Design

A 2-DOF single objective simulated annealing algorithm has been applied to the broadband conceptual design optimization problems in Eqn. 6.27 through Eqn. 6.31. Table 6.10 lists the cooling schedule used by the single objective simulated annealing algorithm. The initial system temperature was set to a value of two and was reduced by a factor of 0.88 after each iteration. Each trial of the algorithm was terminated after 60 iterations.

**TABLE 6.10** Simulated Annealing Cooling Schedule

Cooling Schedule Parameters	Value
Initial System Temperature	2 \$/T1-minute
Temperature Reduction Factor	0.88
Number Iterations	60

### Results

Table 6.11 illustrates the results for the five single objective optimization business case scenarios. In the first scenario, the MDO algorithm converges on a regional architecture that serves 75% of the populated regions of the Earth. Assuming full 1% market penetration, a "Big LEO" system dramatically boosts the system capacity and thus reduces the price per billable T1-minute. In LEO, the same signal-to-noise ratio can be achieved with smaller, less powerful phased array antennas. This in turn reduces the size of the satellites, and thus dramatically reduces the satellite production and launch costs. If the market demand is great enough, adding more satellites is a cost-effective way to add system capacity and boost revenues.

In the second scenario, the MDO algorithm converges on a global architecture that meets the 98% availability requirement over 100% of the populated regions of the Earth. Global coverage is achieved by increasing the inclination of the constellation, leading to an



**TABLE 6.11** Best Business Case Broadband Architectures Found by the MDO Algorithm

Parameter	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
Altitude	LEO	LEO	LEO	LEO	LEO
Inclination (°)	30	75	60	75	75
Satellites/Plane	8	8	4	8	8
# Orbital Planes	10	8	10	6	8
Transmission Power (kW)	2	4	1	4	4
Antenna Area (m <sup>2</sup> )	3.5	5.0	5.0	5.0	5.0
\$/T-1 Minute	0.0023	0.0031	0.0012	0.0026	0.0032
Lifecycle Cost (\$B)	7.43	11.1	3.50	9.0	11.1
Subscriber Years (millions)	25.2	29.5	21.2	29.2	28.0
Market Availability (%)	100	100	95	98	100
Global Availability (%)	75	100	89	100	100

increase in launch costs. In the third scenario, relaxation of the market served satellite availability requirement – enabled by entering into a collaborative agreement with a terrestrial broadband service provider – allows for a smaller constellation which dramatically reduces the lifecycle cost of the system by over a factor of two. This in turn leads to a lower cost per billable T1-minute than in the first two business case scenarios. In the fourth scenario, addition of the global population coverage requirement again increases the size of the constellation, lifecycle cost, and the cost per billable T1-minute. Notice, however, that the cost per billable T1-minute is lower in scenario four than in scenario two, which also mandated global population coverage, due to the addition of a partnering terrestrial broadband service provider that in turn relaxes the satellite availability requirement. In the fifth scenario, increasing the link margin, coverage, and availability requirement for the military increases lifecycle cost, reduces system performance, and results in the highest cost per billable T1-minute architecture.

### Summary

In this section, the broadband architecture conceptual design problem has been modeled as a single objective optimization problem for five separate business case scenarios, and a simulated annealing algorithm has been applied to search the global trade space for the

architectures that minimize the cost per billable T1-minute without completely enumerating the trade space. Different cost-effective architectures were found under each business case scenario. In all five scenarios, however, LEO systems minimized the CPF metric due to their ability to dramatically boost system capacity and performance (i.e. less space loss with a given power-aperture) and reduce lifecycle cost (i.e. smaller, cheaper satellites). These five scenarios illustrate the power of the MMDOSA methodology to search the trade space for different customers and business plans under alternative market scenarios by simply changing the mathematical formulation of the optimization problem – in this case the constraint equations. All five scenarios were analyzed with the same GINA model and single objective simulated annealing algorithm. Thus, the MMDOSA methodology may serve as a powerful, versatile tool for trade space exploration, concept development, and the rapid analysis of alternative commercial space system business plans under uncertain market conditions.

### **6.5.2 Approach 2 – Multiobjective Optimization (2 Dimensions)**

#### **Objective**

In the second stage of the test matrix, the broadband system architecture conceptual design problem is modeled as a multiobjective optimization problem with the intention of finding the Pareto optimal set of architectures without completely enumerating the trade space. This is done by applying the multiobjective multiple solution simulated annealing algorithm developed in Section 3.5.2, which attempts to satisfy simultaneously the competing objectives of minimizing lifecycle cost and maximizing system performance (i.e. the total number of billable T1-minutes or subscriber years provided – system capacity). This second conceptual design problem may be characterized as a multiobjective DSS architecture design problem.

#### **Problem Formulations**

The two-dimensional multiobjective conceptual design optimization problem may be formulated mathematically as:

$$\begin{array}{ll}
\text{Objective:} & \text{Min} \left( \sum_{y=1}^{10} \Phi_y(\Gamma) \right) \text{ AND } \text{Max} \left( \sum_{y=1}^{10} \Psi_y(\Gamma) \right) \\
\text{Constraints:} & \text{Subject to} \\
\text{Isolation} & \text{MAE} \geq 90\% \\
& \text{SNR} \geq 6.4 \text{ dB} \\
& L_M \geq 7.8 \text{ dB} \\
\text{Integrity} & \text{BER} \leq 10^{-5} \\
\text{Rate} & R \geq 1.54 \text{ Mbps per link} \\
\text{Availability} & \epsilon_{\min} \geq 10^\circ \\
& A_{vm} \geq 98\% \\
\text{Bounds} & 0^\circ \leq \gamma_2 \leq 75^\circ \\
& 1 \leq \gamma_3 \leq 8 \\
& 2 \leq \gamma_4 \leq 8 \\
& 1 \text{ kW} \leq \gamma_5 \leq 10 \text{ kW} \\
& 0.5 \text{ m}^2 \leq \gamma_6 \leq 5.0 \text{ m}^2
\end{array} \tag{6.32}$$

where  $y$  is the year in the mission,  $\Phi_y(\Gamma)$  is the cost in year  $y$  of architecture  $\Gamma$ ,  $\Psi_y(\Gamma)$  is the number of billable T1-minutes the architecture provides in year  $y$ , MAE is the multi-access efficiency, SNR is the required T1 link signal-to-noise ratio,  $L_M$  is the link margin to account for rain loss and other losses, BER is the bit error rate,  $R$  is the data transfer rate,  $\epsilon_{\min}$  is the minimum viewing elevation angle required to establish a link between a user terminal and satellite,  $A_{vm}$  is the system availability over the market served, and  $\gamma$ 's are the different elements of the design vector. To evaluate a broadband architecture on the basis of the objective function and constraints in Eqn. 6.32, the design vector is entered into the broadband GINA model. Table 6.2 lists the design vector variables along with the allowable range of values for each of these variables.

In addition to the change in the mathematical formulation, the multiobjective optimization problem also requires a change in the internal mechanics of the simulated annealing algorithm. As explained in Section 3.5,  $\Delta$  is a parameter that is used to determine the probabil-

ity with which the algorithm may move to a less optimal solution with the intent of escaping local minima within the nonconvex trade space. In the single objective case,  $\Delta$  is simply the difference in the objective function value between successive steps of the algorithm. For multiobjective problems, however,  $\Delta$  must take into account the difference in successive steps between all objective functions – that is for all  $N$  dimensions of the problem.

The formula for computing the  $\Delta$  parameter for a multiobjective optimization problem with  $N$  dimensions in the objective function (i.e.  $N$  design goals or  $N$  decision criteria) is:

$$\Delta = \frac{\sum_{n=1}^N \frac{E_n(\Gamma_{i+1}) - E_n(\Gamma_i)}{\delta_n}}{N} \quad (6.33)$$

where  $i$  is the iteration number within the simulated annealing algorithm and  $\delta_n$  is the maximum range in dimension  $n$  observed in the random sample. Depending upon the distribution of the random sample,  $\delta_n$  may be replaced by the  $1\sigma_n$  range,  $2\sigma_n$  range, or  $3\sigma_n$  range for each dimension  $n$  in the objective function.

The  $+2\sigma$  value was used rather than the entire range for the  $\delta_n$  parameter in the computation of  $\Delta$ . The  $+2\sigma$  value for each dimension of the objective function implies that 97.5% of all possible system architectures exhibit a value less than  $+2\sigma$  along that particular dimension. The  $\delta_n$  parameter represents a range around which each objective function will be normalized, and may now be computed as:

$$\delta_n = \bar{x}_n + 2\sigma_n - x_{n_{\min}} \quad (6.34)$$

where  $\bar{x}_n$  is the mean value of metric  $n$ ,  $\sigma_n$  is the sample standard deviation along dimension  $n$ , and  $x_{n_{\min}}$  is the minimum value observed in the random sample along dimension  $n$ . Table 6.7 lists the results for each metric in the objective function in Eqn. 6.32 from the random sample of the broadband trade space. Using this information to solve Eqn. 6.34,

the  $\Delta$  parameter for the multiobjective optimization problem in Eqn. 6.32 may be computed as:

$$\Delta = \frac{\Phi(\Gamma_{i+1}) - \Phi(\Gamma_i)}{\$17.96 \times 10^9} + \frac{\Psi(\Gamma_i) - \Psi(\Gamma_{i+1})}{21.8 \times 10^6 \text{ subscriber years}} \quad (6.35)$$

Table 6.12 lists the cooling schedule used by the single objective simulated annealing algorithm. The initial system temperature was set to one and was reduced by a factor of 0.90 after each iteration. Each trial of the algorithm was terminated after 60 iterations.

**TABLE 6.12** Simulated Annealing Cooling Schedule

Cooling Schedule Parameters	Value
Initial System Temperature	1
Temperature Reduction Factor	0.90
Number Iterations	60

## Results

Figure 6.22 illustrates the Pareto optimal set of broadband architectures within the global trade space found by five cumulative trials of the multiobjective, multiple solution simulated annealing algorithm; each of which found Pareto optimal architectures after evaluating only 0.14% of the global trade space. It should be pointed out that the diagonal iso-cost per T1-minute lines are different from the actual charged cost per T1-minute metric computed in Eqn. 6.25 and used in Section 6.5.1. This is because the lifecycle cost on the y-axis in Figure 6.22 is not discounted to account for the time value of money. The chart in Figure 6.23 lists the Pareto optimal broadband architectures identified by the MDO algorithm. Within this Pareto optimal set, the systems engineer cannot improve the performance of the broadband architecture without also increasing lifecycle cost. Notice that several families of architectures inhabit the Pareto boundary, with LEO and MEO architectures occupying the low-cost, low-performance end of the boundary, and LEO architectures occupying the medium through high cost and performance range of the boundary.

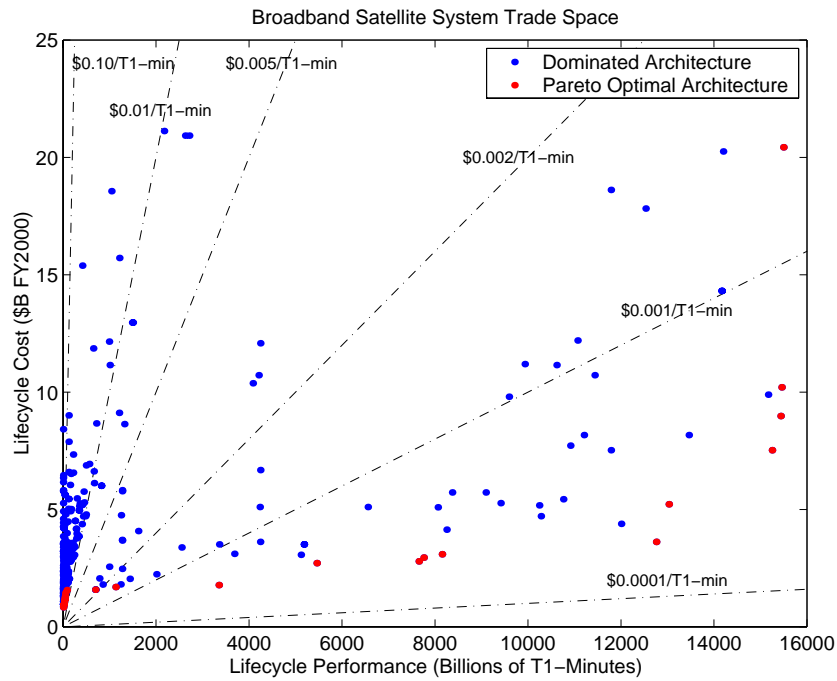


Figure 6.22 Broadband Trade Space Pareto Boundary

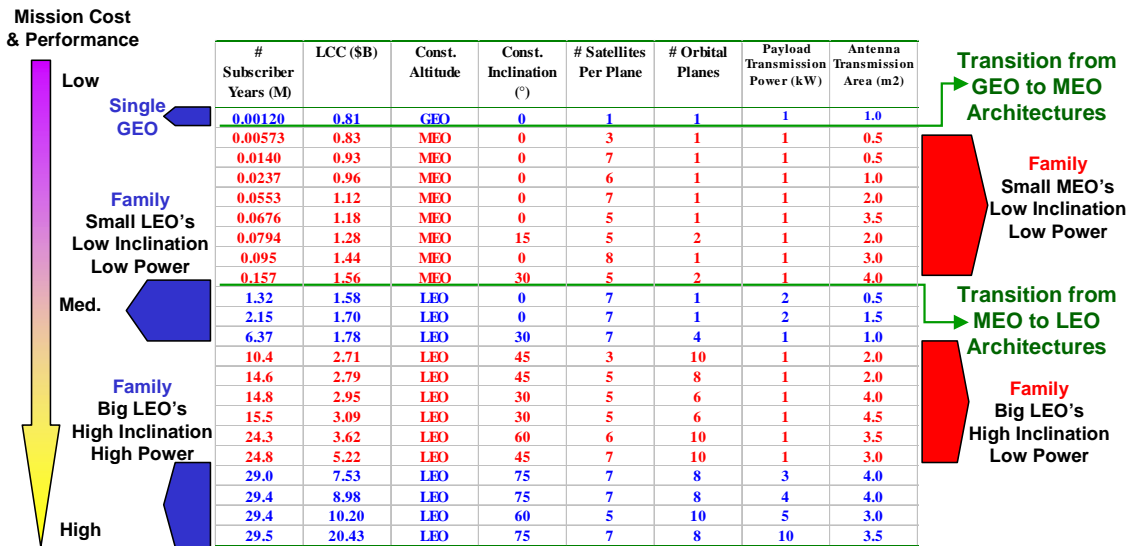


Figure 6.23 Pareto Optimal Architectures Found by the MDO Algorithm

## Summary

In this section, the broadband architecture conceptual design problem has been modeled as a two-dimensional multiobjective optimization problem, and the multiobjective multiple solution simulated annealing algorithm has been applied to search the broadband global trade space for the Pareto optimal set of system architectures without completely enumerating the trade space. Several families of Pareto optimal architectures were found ranging from low-cost, low-performance designs to high-cost, high-performance designs. Interestingly, there was a smooth transition from GEO architectures at the low-cost, low-performance end of the Pareto optimal set through MEO architectures and finally LEO architectures at the high-cost, high-performance end of the Pareto optimal set.

Risk is typically defined as the product of the probability of failure and the cost of failure [Browning, 1998]. For the broadband mission, the cost of failure is equal to the present value of the cost to develop and deploy the system – the present value of the amount of money invested into the system. The probability of failure is a function of the number of subscribers that must be secured to ensure a profitable operation. As the required number of subscribers increases, so does the probability of failure. This implies that LEO systems have a higher probability of failure than MEO and GEO systems, MEO systems have a higher probability of failure than GEO systems, and GEO systems have the lowest probability of failure. Even assuming that the probability of failure remains constant across all orbital altitudes, these results indicate that geosynchronous architectures provide a low-risk solution (i.e. smaller initial investment = lower cost of failure) to the broadband satellite communications conceptual design problem, but with a limited maximum return on investment. Conversely, low Earth orbit constellations represent an alternative high-risk solution (i.e. larger initial investment = higher cost of failure) to the same conceptual design problem. LEO constellations require a much greater up front investment, but provide the capacity for a greater return on investment as well. The actual return on investment of a system with such a large capacity will depend upon the size of the broadband market and the ability of satellite services to penetrate this market – both of which are

highly uncertain variables in an uncertain marketplace. The most successful satellite broadband systems will be those that properly match their system capacity to the market demand.

### **6.5.3 Approach 3 – Multiobjective Optimization (4 Dimensions)**

#### **Objective**

In the third stage of the test matrix, the broadband system architecture conceptual design problem is modeled as a multiobjective optimization problem with the intention of finding the Pareto optimal set of architectures without completely enumerating the trade space. This is done by applying the multiobjective multiple solution simulated annealing algorithm developed in Chapter 3.5.2, which attempts to satisfy simultaneously the competing objectives of minimizing lifecycle cost, maximizing system performance, maximizing the number of satellites simultaneously in view over the market served, and maximizing the system's global availability. This third conceptual design problem may be characterized as a multiobjective DSS architecture design problem.

#### **Problem Formulation**

This four-dimensional multiobjective conceptual design optimization problem may be formulated mathematically as:



$$\begin{aligned}
 \text{Objective:} \quad & \text{Min} \left( \sum_{y=1}^{10} \Phi_y(\Gamma) \right) \text{ AND } \text{Max} \left( \sum_{y=1}^{10} \Psi_y(\Gamma) \right) \\
 & \text{AND } \text{Max} \frac{\sum_{i=1}^{t_s} \sum_{j=1}^{g_r} \text{SIV}_{ij}(\Gamma)}{g_r} \text{ AND } \text{Max} \frac{\sum_{j=1}^{g_r} \sum_{i=1}^{t_s} \text{COV}_{ji}(\Gamma)}{t_s} \\
 \text{Constraints:} \quad & \text{Subject to} \\
 \text{Isolation} \quad & \text{MAE} \geq 90\% \\
 & \text{SNR} \geq 6.4 \text{ dB} \\
 & L_M \geq 7.8 \text{ dB} \\
 \text{Integrity} \quad & \text{BER} \leq 10^{-5} \\
 \text{Rate} \quad & R \geq 1.54 \text{ Mbps per link} \\
 \text{Availability} \quad & \epsilon_{\min} \geq 10^\circ \\
 & A_{vm} \geq 98\% \\
 \text{Bounds} \quad & 0^\circ \leq \gamma_2 \leq 75^\circ \\
 & 1 \leq \gamma_3 \leq 8 \\
 & 2 \leq \gamma_4 \leq 8 \\
 & 1 \text{ kW} \leq \gamma_5 \leq 10 \text{ kW} \\
 & 0.5 \text{ m}^2 \leq \gamma_6 \leq 5.0 \text{ m}
 \end{aligned} \tag{6.36}$$

where  $y$  is the year in the mission,  $\Phi_y(\Gamma)$  is the cost in year  $y$  of architecture  $\Gamma$ ,  $\Psi_y(\Gamma)$  is the number of billable T1-minutes the architecture provides in year  $y$ ,  $i$  represents a specific time step in the simulation,  $t_s$  is the total number of time steps in the simulation,  $j$  represents a particular Earth grid point with a user terminal,  $g_r$  is the total number of Earth grid points as defined by the Earth grid resolution in the constants vector,  $\text{SIV}_{ij}$  is the number of satellites in view of gridpoint  $j$  at time step  $i$ ,  $\text{COV}_{ji}$  is a binary variable denoting whether or not Earth grid point  $j$  is covered during time step  $i$ , MAE is the multi-access efficiency, SNR is the required T1 link signal-to-noise ratio,  $L_M$  is the link margin to account for rain loss and other losses, BER is the bit error rate,  $R$  is the data transfer rate,

$\varepsilon_{\min}$  is the minimum viewing elevation angle to establish a link between a user terminal and satellite,  $A_{vm}$  is the system availability over the market served, and  $\gamma$ 's are the different elements of the design vector. To evaluate a broadband system architecture on the basis of the objective function and constraints in Eqn. 6.36, the design vector is entered into the broadband GINA model. Table 6.2 lists the design vector variables along with the allowable range of values for each of these variables.

As previously discussed, formulating the broadband design optimization problem as a multiobjective problem requires a change in the internal mechanics of the simulated annealing algorithm. For multiobjective problems,  $\Delta$  must take into account the difference in successive steps between all objective functions – that is for all  $N$  dimensions of the problem. Eqn. 6.33 gives the formula for computing the  $\Delta$  parameter for a multiobjective optimization problem with multiple dimensions in the objective function.

The  $+2\sigma$  value was used for the  $\delta_n$  parameter in the computation of  $\Delta$ . Using the information learned from the random sample in Section 6.4, the  $+2\sigma$  values for  $\delta_n$  are computed in Table 6.13 according to Eqn. 6.34.

**TABLE 6.13**  $\delta_n$  Values for the Computation of  $\Delta$

Metric	$\delta_n = \bar{x}_n + 2\sigma_n - x_{n_{\min}}$
Lifecycle Cost (\$B)	17.96
Performance (subscriber years)	$21.8 \times 10^6$
Mean # Satellites in View Over Market Served	15.3
Global Availability	1

Using this information to solve Eqn. 6.33, the  $\Delta$  parameter for the multiobjective optimization problem in Eqn. 6.36 may be computed as:

$$\Delta = \frac{\frac{\Phi(\Gamma_{i+1}) - \Phi(\Gamma_i)}{\$17.96 \times 10^9} + \frac{\Psi(\Gamma_i) - \Psi(\Gamma_{i+1})}{21.8 \times 10^6 \text{ sub. years}} + \frac{\text{SIV}(\Gamma_i) - \text{SIV}(\Gamma_{i+1})}{15.3} + \frac{\text{COV}(\Gamma_i) - \text{COV}(\Gamma_{i+1})}{1}}{4} \quad (6.37)$$

Table 6.12 lists the cooling schedule used by the single objective simulated annealing algorithm. The initial system temperature was set to a value of one and was reduced by a factor of 0.98 after each iteration. Each trial of the algorithm was terminated after 180 iterations.

**TABLE 6.14** Simulated Annealing Cooling Schedule

Cooling Schedule Parameters	Value
Initial System Temperature	1
Temperature Reduction Factor	0.98
Number Iterations	180

## Results

After four separate runs of the multiobjective multiple solution simulated annealing algorithm, 81 Pareto optimal architectures were identified. The Pareto optimal architectures are listed in Table 6.15. Within this Pareto optimal set, the systems engineer cannot simultaneously improve all four metrics of interest. These results illustrate how higher-order (i.e. more than two dimensions) Pareto optimal sets can become very large very quickly, resulting in the identification of more Pareto optimal architectures than can be carried forward into the next phase of the design and development process. When this occurs, the size of the Pareto optimal set may be reduced by placing constraints on the allowable value for each metric. For example, the systems engineer may wish to consider only those architectures with a lifecycle cost of less than \$2 B, a mean number of satellites in view over markets served of greater than two, etc. Another method by which to distinguish and rank architectures within a large Pareto optimal set is by multi-attribute utility function theory [de Neufville, 1990].

## Summary

In this section, the broadband architecture conceptual design problem has been modeled as a four-dimensional multiobjective optimization problem, and the multiobjective multiple solution simulated annealing algorithm has been applied to search the broadband global

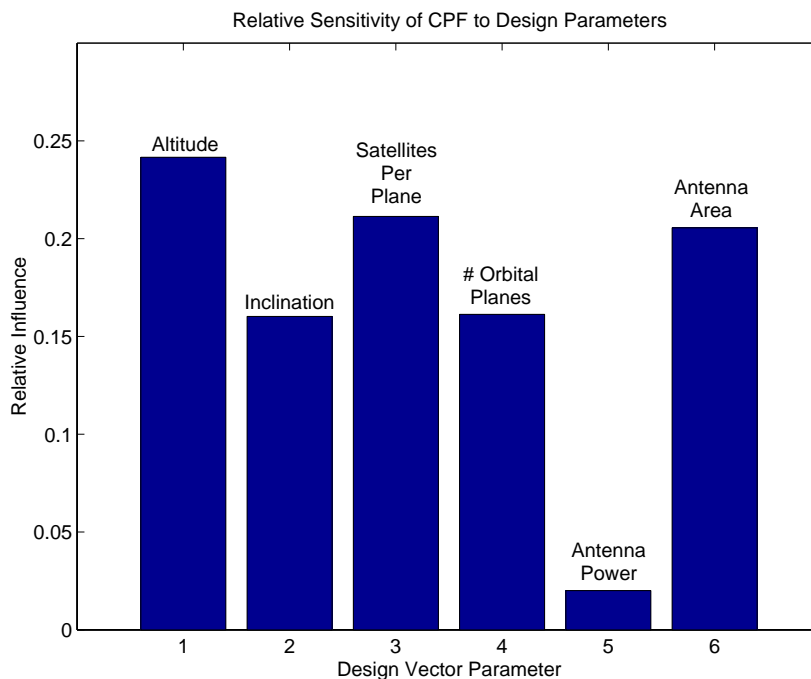
TABLE 6.15 Four-Dimensional Pareto Optimal Set

LCC (\$B)	# Subscr. Year (M)	Mean # SIV	Global Avail.	Const. Altitude	Const. Inc. (°)	# Sat.Per Plane	# Orbital Planes	Power (kW)	Antenna Area-m <sup>2</sup>
0.81	0.0012	1.0	0.28	GEO	0	1	1	1	1.0
0.84	0.0018	1.0	0.28	GEO	0	1	1	1	1.5
0.86	0.0025	1.0	0.28	GEO	0	1	1	1	2.0
0.87	0.0037	1.0	0.63	GEO	0	2	1	1	1.5
0.92	0.0049	1.0	0.63	GEO	0	2	1	1	2.0
0.95	0.0062	1.0	0.63	GEO	0	2	1	1	2.5
0.95	0.0076	1.1	0.84	GEO	0	3	1	1	2.0
0.98	0.0096	1.8	0.89	GEO	0	5	1	1	1.5
1.00	0.0091	2.4	0.90	GEO	0	7	1	1	1.0
1.06	0.0137	2.4	0.90	GEO	0	7	1	1	1.5
1.09	0.0152	1.4	0.87	GEO	0	4	1	1	3.0
1.12	0.0182	2.4	0.90	GEO	0	7	1	1	2.0
1.12	1.89	1.0	0.34	LEO	0	7	1	1	2.0
1.13	0.0194	2.0	0.90	GEO	0	6	1	1	2.5
1.13	0.0354	2.0	0.86	MEO	15	3	2	1	1.5
1.17	0.0202	1.4	0.87	GEO	0	4	1	1	4.0
1.19	0.0459	1.8	1.00	MEO	45	3	2	1	2.0
1.25	0.0232	2.0	0.90	GEO	0	6	1	1	3.0
1.28	0.0794	2.9	0.90	MEO	15	5	2	1	2.0
1.47	0.0645	4.5	1.00	MEO	30	4	4	1	1.0
1.47	0.0637	4.9	1.00	MEO	75	4	4	1	1.0
1.56	0.1573	2.8	1.00	MEO	30	5	2	1	4.0
1.62	2.35	1.0	0.34	LEO	0	7	1	1	5.0
1.88	0.164	2.4	0.75	MEO	0	7	1	2	3.0
1.99	2.41	1.0	0.34	LEO	0	7	1	2	4.0
2.02	0.123	8.5	1.00	MEO	30	5	6	1	1.0
2.05	2.42	1.0	0.34	LEO	0	7	1	2	4.5
2.11	2.42	1.0	0.34	LEO	0	7	1	2	5.0
2.11	4.97	2.9	0.95	LEO	75	6	6	1	1.0
2.17	0.287	4.5	1.00	MEO	30	4	4	1	4.5
2.21	6.91	2.4	0.68	LEO	30	4	8	1	1.0
2.40	0.121	9.2	1.00	MEO	75	3	10	1	1.0
2.45	6.48	5.8	0.56	LEO	15	7	8	1	1.0
2.49	0.129	9.9	1.00	MEO	75	8	4	1	1.0
2.55	0.325	3.7	1.00	MEO	75	3	4	2	3.5
2.65	0.38	4.5	0.98	MEO	75	7	2	2	3.5
2.71	10.38	2.1	0.77	LEO	45	3	10	1	2.0
2.83	0.44	10.1	1.00	MEO	30	6	6	1	3.0
3.20	0.57	3.4	1.00	MEO	30	6	2	3	4.0
3.22	0.48	7.4	1.00	MEO	75	3	8	1	5.0
3.39	0.58	9.0	1.00	MEO	30	4	8	1	4.5
3.73	0.76	4.5	1.00	MEO	30	8	2	3	4.0
3.86	0.64	4.8	0.97	MEO	45	2	8	2	5.0
4.49	0.89	9.9	1.00	MEO	75	8	4	2	3.5
4.53	23.74	2.5	0.89	LEO	60	4	10	2	4.0
4.76	10.36	6.4	0.59	LEO	15	8	10	1	1.5
4.78	0.94	3.4	1.00	MEO	30	6	2	5	4.0
4.94	1.18	6.1	1.00	MEO	60	5	4	3	5.0
5.04	26.48	3.2	0.97	LEO	60	7	8	1	5.0
5.04	1.13	16.5	0.90	MEO	15	7	8	1	5.0
5.25	1.29	6.8	1.00	MEO	30	6	4	3	4.5
5.27	1.16	9.0	1.00	MEO	30	4	8	2	4.5
5.49	19.80	3.3	0.99	LEO	75	6	8	2	3.0
5.86	1.29	7.1	0.90	MEO	15	3	8	3	4.5
6.79	1.57	6.1	1.00	MEO	60	5	4	5	4.0
7.20	22.30	4.2	0.73	LEO	30	7	8	3	3.0
7.22	1.59	6.0	0.97	MEO	45	2	10	4	5.0
7.37	2.56	14.6	1.00	MEO	60	6	8	3	4.5
7.37	2.58	14.1	0.90	MEO	15	6	8	3	4.5
8.11	22.42	3.8	0.71	LEO	30	5	10	3	5.0
8.13	28.21	4.3	0.88	LEO	45	7	10	3	2.0
8.60	2.68	13.5	1.00	MEO	30	8	6	4	3.5
8.74	29.19	2.4	0.97	LEO	60	7	6	4	3.5
8.76	22.42	4.5	0.74	LEO	30	6	10	3	5.0
8.89	29.49	4.0	0.97	LEO	60	7	10	3	4.0
10.13	2.83	9.4	0.90	MEO	15	4	8	5	4.5
11.02	21.87	3.6	0.98	LEO	60	8	8	5	0.5
11.30	27.06	3.4	0.99	LEO	75	8	6	6	4.5
11.57	4.72	15.2	1.00	MEO	60	5	10	5	5.0
12.00	29.49	4.0	1.00	LEO	75	8	8	5	3.0
13.25	27.56	4.4	1.00	LEO	75	7	10	6	1.0
13.36	1.55	18.8	0.90	MEO	15	8	8	6	1.0
13.47	22.42	4.8	0.73	LEO	30	8	8	6	3.0
13.47	29.49	4.0	1.00	LEO	75	8	8	6	3.0
13.67	3.97	21.5	1.00	MEO	75	7	10	5	3.0
13.96	0.82	24.3	1.00	MEO	60	8	10	5	0.5
14.01	26.79	4.5	0.75	LEO	30	7	10	6	3.0
15.98	29.49	5.0	1.00	LEO	75	8	10	6	3.0
16.98	6.17	19.7	1.00	MEO	75	8	8	6	4.5
24.86	8.51	16.5	0.90	MEO	15	7	8	9	4.5
27.45	9.62	18.8	0.90	MEO	15	8	8	9	4.5

trade space for the Pareto optimal set of system architectures without completely enumerating the trade space. A large Pareto optimal set of architectures was found ranging from low-cost, low-performance, low-availability designs to high-cost, high-performance, high-availability designs. This Pareto optimal set represents the best set of architectures with which the systems engineer can simultaneously minimize lifecycle cost, maximize system capacity, maximize the number of satellites in view over the market served, and maximize global availability.

## 6.6 MMDOSA Step 5 – Interpret Results (Sensitivity Analysis)

Figure 6.24 illustrates the ANOVA results for the broadband trade space. As one can see, constellation altitude exerts the greatest relative influence on the cost per billable T1-minute metric. These results correspond with those in Figure 6.23, where a direct correlation between constellation altitude, financial risk as measured by the investment (i.e. lifecycle cost) required to field the system, and the potential financial reward that results if the



**Figure 6.24** ANOVA Results for the Broadband Trade Space

market utilizes the full system capacity was identified. ANOVA provides the systems engineer with information on which design parameters give the most leverage over various aspects of the design. In this case, the selection of the constellation altitude is a key decision that goes a long way in determining the range of the cost per billable T1-minute the system must charge to achieve a zero net present value (i.e. a 30% internal rate of return) after 10 years of operation.

## **6.7 MMDOSA Step 6 – Iterate**

Multiple trials from different randomly selected starting points were executed for both the single objective and multiobjective optimization problems. The cumulative results were analyzed and used to warm start successive trials.

## **6.8 MMDOSA Step 7 – Converge on Best System Architectures**

Having applied the first six steps of the MMDOSA methodology to the broadband conceptual design problem, the best architectures for the commercial organization to carry forward into the next program phase, Phase B, may be recommended. Architectures are recommended for each of the following three design scenarios:

1. The commercial organization wishes to select the most cost-effective broadband satellite system architecture as measured by the cost per billable T1-minute metric.
2. The commercial organization wishes to select a broadband satellite system architecture that simultaneously minimizes lifecycle cost and maximizes system capacity.
3. The commercial organization wishes to select a broadband satellite system architecture that simultaneously minimizes lifecycle cost, maximizes system capacity, maximizes the number of satellites simultaneously in view over the market served, and maximizes the system's global availability.

### **6.8.1 Recommended Most Cost-Effective Architectures**

As illustrated in Section 6.5.1, the composition of the most cost-effective architectures varies as a function of the problem formulation constraints dictated by the business plan

according to which the system will be deployed. For example, mandating global coverage adds the mathematical constraint that the system availability requirement be met over the entire surface of the Earth, making regional service architectures infeasible design solutions and eliminating these potentially more cost-effective architectures from consideration. However, the minimum cost per billable T1-minute architectures found by the single objective simulated annealing algorithm in all five business case scenarios do share some common traits as illustrated in Table 6.16. In each business case, the most cost-effective – assuming the market absorbs the full system capacity – broadband satellite architectures utilize low power, large aperture satellites located in low Earth, medium to high inclination orbits.

**TABLE 6.16** Recommended Family of Architectures for Phase B

<b>Design Vector</b>	<b>Family</b>
Constellation Altitude	LEO
Constellation Inclination	30° - 75°
# Satellites Per Plane	4 - 8
# Orbital Planes	6 - 10
Payload Power	1 kW - 4 kW
Antenna Transmission Area	3.5 m <sup>2</sup> - 5.0 m <sup>2</sup>

## 6.8.2 Recommended 2-Dimensional Pareto Optimal Architectures

Figure 6.23 lists the complete set of two-dimensional Pareto optimal architectures found by the multiobjective multiple solution simulated annealing algorithm within the global trade space. This set of architectures defines a boundary along which the systems engineer cannot improve system performance (i.e. system capacity) without also increasing lifecycle cost. Table 6.17 lists the recommended Pareto optimal broadband satellite architectures as a function of the size of the commercial investment. As the initial investment increases (i.e. the investors take on more risk assuming that either the probability of failure is constant independent of the system capacity or increases as capacity increases), greater levels of system capacity may be achieved. If the market absorbs this capacity,

then these architectures will provide the greatest return on investment. Remembering that risk is defined as the cost of failure times the probability of failure, there is a direct correlation between financial risk and potential financial reward. These are the architectures which should be carried forward into Phase B of the broadband satellite program if the commercial organization wishes to select an architecture that simultaneously minimizes lifecycle cost and maximizes the number of billable T1-minutes provided.

**TABLE 6.17** Recommended Broadband Satellite Architectures as a Function of Financial Investment

Parameter	\$1.0 B Investment	\$2.0 B Investment	\$5.0 B Investment	\$10.0 B Investment
Constellation Altitude (km)	MEO	LEO	LEO	LEO
Constellation Inclination (°)	0	30	60	75
# Satellites Per Plane	6	7	6	7
# Orbital Planes	1	4	10	8
Payload Power (kW)	1	1	1	4
Antenna Transmission Area (m <sup>2</sup> )	1.0	1.0	3.5	4.0
Lifecycle Cost (\$B FY2000)	0.96	1.78	3.62	8.98
# Subscriber Years (millions)	0.0237	6.37	24.3	29.4
\$/Billable T-1 Minute	0.19	0.0016	0.0011	0.0025

### 6.8.3 Recommended 4-Dimensional Pareto Optimal Architectures

Table 6.15 lists the complete set of four-dimensional Pareto optimal architectures found by the multiobjective, multiple solution simulated annealing algorithm within the global trade space. This set of architectures defines a boundary along which the systems engineer cannot simultaneously reduce lifecycle cost, increase system capacity, increase the average number of satellites simultaneously in view over the market served, and increase global availability. Table 6.18 lists in order of increasing lifecycle cost the Pareto optimal broadband architectures with a lifecycle cost less than \$3 billion, a system capacity greater than 100,000 subscriber years, an average number of satellites in view over the market served of three or more, and a global availability of 99% or better.



These are the architectures which should be carried forward into Phase B of the broadband satellite program if the commercial organization wishes to select an architecture that simultaneously minimizes lifecycle cost, maximizes system capacity, maximizes the average number of satellites simultaneously in view over the market served, and maximizes global availability.

**TABLE 6.18** Recommended Four-Dimensional Pareto Optimal Broadband Satellite Architectures

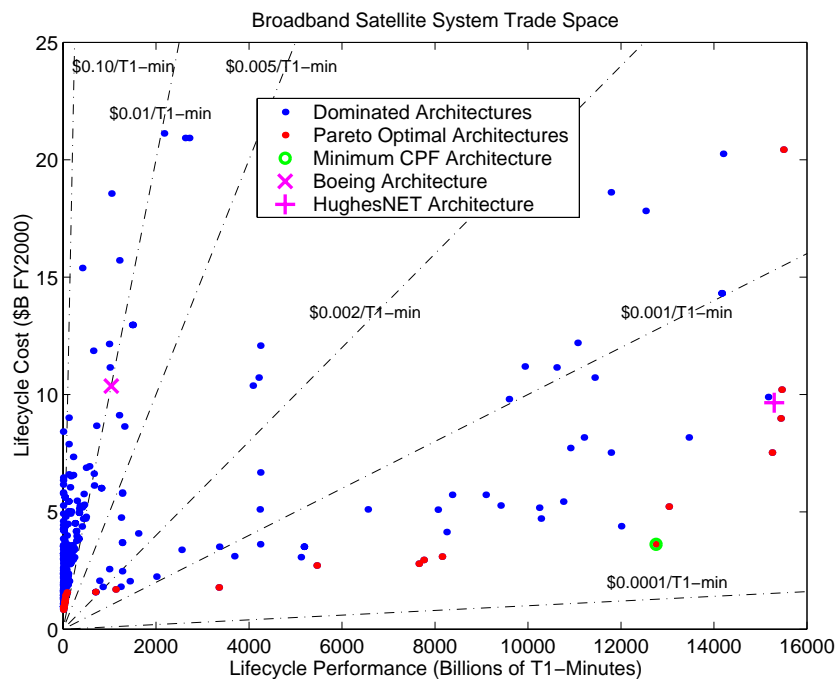
<b>Parameter</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
Constellation Altitude (km)	MEO	MEO	MEO	MEO	MEO	MEO
Constellation Inclination (°)	30	30	75	75	75	30
# Satellites Per Plane	5	4	3	8	3	6
# Orbital Planes	6	4	10	4	4	6
Payload Power (kW)	1	1	1	1	2	1
Antenna Transmission Area (m <sup>2</sup> )	1.0	4.5	1.0	1.0	3.5	3.0
Lifecycle Cost (\$B FY2000)	2.02	2.17	2.40	2.49	2.55	2.83
# Subscriber Years (thousands)	123	287	121	129	325	439
Mean # Satellites in View	8.5	4.5	9.2	9.9	3.7	10.1
Global Availability (Populated) (%)	100%	100%	100%	100%	100%	100%
\$/Billable T-1 Minute	0.087	0.042	0.11	0.10	0.045	0.036

## 6.9 Conclusions

This chapter has illustrated the application of the MMDOSA methodology to the conceptual design of a commercial broadband satellite communications mission. First, the GINA methodology was applied to create a GINA simulation model for the system. Next, univariate studies and random sampling were executed to obtain initial information on the trade space. With this information in hand, single objective and multiobjective simulated annealing algorithms were applied to search for the most cost-effective architectures and the Pareto optimal architectures within the trade space. After a sensitivity analysis, specific system architectures were identified and recommended for further detailed study in the next phase of the commercial broadband satellite communications program.

Figure 6.25 illustrates the broadband trade space, including the Pareto optimal set of architectures, the most cost-effective minimum cost per billable T1-minute architecture, and point design architectures developed by The Boeing Corporation and Hughes Satellite Systems (i.e. now Boeing Satellite Systems) as documented in their respective FCC filings [Boeing, 1999; Hughes Communications, 1999]. Table 6.19 and Table 6.20 compare the lifecycle cost, system capacity, and system cost per billable T1-minute as computed by the GINA model for the industry point design architectures with the following three best broadband satellite architectures found by the MMDOSA methodology:

1. The most cost-effective architecture (i.e. minimizes the CPF metric).
2. The Pareto optimal architecture that provides equivalent system capacity at a reduced lifecycle cost.
3. The Pareto optimal architecture that provides improved system capacity for an equivalent lifecycle cost.



**Figure 6.25** Broadband Satellite System Trade Space

Table 6.19 compares The Boeing Corporation point design with the best designs found by the MMDOSA methodology. The Boeing architecture places twenty 8.4 KW, 3.0 m<sup>2</sup> aperture satellites into a four orbital plane, 57° inclination medium Earth orbit constellation [Boeing, 1999]. The minimum cost per billable T1-minute architecture reduces the cost per billable T1-minute of the system by 97%. The Pareto optimal equivalent performance architecture reduces lifecycle cost by 85%, and the Pareto optimal equivalent lifecycle cost architecture improves system performance by 1392%. In each instance, the MMDOSA methodology identified better, more cost-effective broadband satellite architectures than the Boeing point design.

**TABLE 6.19** Comparison of Boeing Point Design with Designs Found by the MMDOSA Methodology

Architecture	Boeing Reference Design	MMDOSA Minimum \$/T1-Minute Architecture	Percent Improvement	Pareto Optimal Equivalent Performance	Percent Improvement	Pareto Optimal Equivalent LCC	Percent Improvement
Constellation Altitude	MEO	LEO		LEO		LEO	
Constellation Inclination(°)	<b>57</b>	60		0		60	
# Satellites Per Plane	<b>5</b>	6		7		5	
# Orbital Planes	<b>4</b>	10		1		10	
Payload Power (kW)	<b>8.4</b>	1.0		2.0		5.0	
Antenna Trans. Area (m <sup>2</sup> )	<b>3.0</b>	3.5		0.5		3.0	
LCC (\$B)	<b>10.36</b>	3.62	65%	1.58	<b>85%</b>	10.20	2%
# Subscriber Years (M)	<b>1.97</b>	24.3	1134%	1.32	-33%	29.4	<b>1392%</b>
CPF (\$/T1-Minute)	<b>0.032</b>	0.0011	<b>97%</b>	0.0076	76%	0.0029	91%

Table 6.20 compares the HughesNET point design with the best designs found by the MMDOSA methodology. The HughesNET architecture places seventy 4.0 KW, 1.5 m<sup>2</sup> aperture satellites into a ten orbital plane, 55° inclination low Earth orbit constellation [Hughes Communications, 1999]. The minimum cost per billable T1-minute architecture reduces the cost per billable T1-minute of the system by 59%. The Pareto optimal equivalent performance architecture reduces lifecycle cost by 22%, and the Pareto optimal equivalent lifecycle cost architecture improves system performance by 1%. In each instance, the MMDOSA methodology identified better, more cost-effective broadband satellite architectures than the HughesNET point design.

**TABLE 6.20** Comparison of HughesNET Point Design with Designs Found by the MMDOSA Methodology

Architecture	HughesNET Reference Design	MMDOSA Minimum \$/T1-Minute Architecture	Percent Improvement	Pareto Optimal Equivalent Performance	Percent Improvement	Pareto Optimal Equivalent LCC	Percent Improvement
Constellation Altitude (km)	LEO	LEO		LEO		LEO	
Constellation Inclination(°)	55	60		75		75	
# Satellites Per Plane	7	6		7		7	
# Orbital Planes	10	10		8		8	
Payload Power (kW)	4	1.0		3.0		4.0	
Antenna Trans. Area (m <sup>2</sup> )	1.5	3.5		4.0.0		4.0	
LCC (\$B)	9.65	3.62	62%	7.53	22%	8.98	7%
# Subscriber Years (M)	29.10	24.3	-16%	29.0	-0.3%	29.4	1%
CPF (\$/T1-Minute)	0.0027	0.0011	59%	0.0021	22%	0.0025	7%

As one can see, the MMDOSA methodology enabled the systems engineer to identify better, more cost-effective, greater revenue generating architectures than those previously considered by industry. Based on these results, the HughesNET architecture – which is nearly Pareto optimal – is more cost-effective than the Boeing architecture. For a slightly smaller initial investment, the HughesNET architecture provides a fourteen-fold increase in system capacity. Further, an interesting correlation between orbit selection, financial investment risk, and potential financial reward was identified. In this manner, the MMDOSA methodology improves the quality of the work performed by systems engineers during the conceptual design phase of a commercial DSS program.

# Chapter 7

## CONCLUSIONS AND RECOMMENDATIONS

### 7.1 Thesis Summary

This thesis develops a methodology – the multiobjective, multidisciplinary design optimization systems architecting (MMDOSA) methodology for the conceptual design of distributed satellite systems – for mathematically modeling distributed satellite systems as optimization problems. An increasing number of space missions are utilizing DSS architectures in which multiple satellites work in a coordinated fashion to achieve improvements in performance, cost, and survivability [Shaw et al, 2000]. The trade space for distributed satellite systems can be enormous – too large to enumerate, analyze, and compare all possible system architectures. The MMDOSA methodology enables an efficient search of the trade space for the best families of distributed satellite system (DSS) design solutions and explores system architectures that might not otherwise be considered during the conceptual design phase. In doing so, the methodology adds structure to the conceptual design process and advances the state-of-the-art in complex distributed satellite network design by providing a means to explore the trade space before converging on a single point design. Historically, the conceptual design of space systems has been unstructured, with designers often pursuing a single concept or modifying an existing idea rather than generating new alternatives [Mosher, 1998]. With this traditional aerospace design approach, there is no guarantee that a system-level focus will be taken, and often the final system architecture chosen achieves only feasibility instead of optimality [Mosher, 1996].

System-level trades are often delayed until after a point design has been selected because of the perceived time and effort required to conduct a credible analysis [Riddle, 1998]. As a result, the final design is often inefficient, leaving room for significant improvements in performance and lifecycle cost. By not properly exploring the system trade space and converging upon an efficient or even optimal solution during the conceptual design phase, the lifecycle cost of the system can greatly increase as modifications are required to properly integrate and operate the system during the latter stages of the design process, when changes become more expensive to implement [Shishko & Chamberlain, 1995]. According to the International Council on Systems Engineering (INCOSE), 70%-90% of the development cost of a large system is predetermined after only 5%-10% of the development time has been completed [INCOSE, 1998]. The ability to use multidisciplinary design optimization (MDO) algorithms up front in the conceptual design phase via MMDOSA will give systems engineers greater power to control lifecycle costs.

Chapter 1 motivates the need for MMDOSA in greater detail and includes a review of previous work in the field. Chapter 2 discusses how MMDOSA fits into the space systems design and development process, defines the mathematical optimization foundation on which the methodology is based, and formulates the DSS conceptual design problem as an optimization problem. Chapter 3, the centerpiece of the thesis, details the MMDOSA methodology step by step. The seven step methodology is summarized below.

1. Create the GINA model.
  - Define the mission objective and conceptual design phase objective.
  - Transform the space system into an information network.
  - Develop system metrics.
  - Partition the conceptual design problem.
  - Develop the simulation software.
2. Perform univariate studies.
  - Vary one parameter within a baseline system architecture along a range of values and measure how the system attributes change.
  - Develop an initial feel for the DSS local trade space.
  - Identify model fidelity problems.
3. Take a random sample of the trade space.
  - Obtain a glimpse of the global trade space.

- Identify initial bounds for each optimization metric.
  - Obtain statistical data to tailor MDO algorithms.
4. Formulate and apply MDO algorithms.
    - Apply the single objective optimization simulated annealing algorithm to identify the best family(s) of system architectures on the basis of the metric of interest.
    - Apply the multiobjective optimization simulated annealing algorithms to identify the Pareto optimal set of system architectures with respect to all of the decision criteria.
  5. Interpret results (sensitivity analysis).
    - Identify the most influential design variables.
    - Identify the most important high fidelity models.
  6. Iterate.
    - Increase the fidelity of critical models.
    - Modify the simulated annealing algorithm cooling schedule.
    - Change the simulated annealing algorithm DOF parameter.
    - Warm start a new optimization run.
    - Run additional trials of the same optimization algorithm.
  7. Converge on system architectures to be focused on in the next phase of the design process.
    - Identify the best family(s) of system architectures from the single objective optimization.
    - Identify the Pareto optimal set of system architectures from the multiobjective optimization.

Of the four classes of MDO techniques – Taguchi methods, heuristics, gradient methods, and univariate methods – tested on the single objective optimization DSS conceptual design problem, the heuristic simulated annealing algorithm found the best system architectures with the greatest consistency due to its ability to escape local optima within a non-convex trade space. Accordingly, the simulated annealing algorithm became the core single objective MDO algorithm within the MMDOSA methodology. Next, the problem scope was greatly increased by expanding from single objective optimization problems to multiobjective optimization problems. This allowed more accurate modeling and capturing of the trades involved in real-world DSS conceptual design problems. Both an MDO algorithm that searches for a single Pareto optimal architecture and an MDO algorithm that searches for multiple Pareto optimal architectures during a single run – the multiob-

jective single solution simulated annealing algorithm and the multiobjective multiple solution simulated annealing algorithm – were developed. Lastly, the utility in knowing the global Pareto boundary of a DSS design trade space was discussed, and several methods were explored for approximating this global Pareto boundary with only a limited knowledge of the full DSS design trade space. The Pareto boundary empowers the systems engineer to capture trades between any desired design attributes along the most efficient set of conceptual design solutions for a DSS. Within this Pareto optimal set, the systems engineer cannot simultaneously improve all of the metrics of interest when choosing one Pareto optimal architecture over another. Finally, methods for improving the performance of the simulated annealing algorithm were explored, and it was found that the 2-DOF variant of the simulated annealing algorithm was the most effective at both single objective and multiobjective searches of a DSS trade space.

The versatility of the MMDOSA methodology was demonstrated by applying it to case studies representing the three most common classes of DSS conceptual design problems – civil, defense/national security, and commercial missions. In the first case study presented in Chapter 4, the NASA Origins Terrestrial Planet Finder civil mission, MMDOSA:

1. Found architectures that provide higher levels of performance (i.e. more images) for lower lifecycle costs than any of the proposed industry designs.
2. Determined that structurally connected architectures are more cost-effective than separated spacecraft architectures for the planet detection and atmosphere characterization mission.
3. Identified the performance limit of structurally connected architectures beyond which separated spacecraft architectures are the only feasible solution.

All five families of TPF architectures within the most cost-effective region of the trade space use 4 m diameter apertures and a structurally connected interferometer. These families span a wide range of orbits beyond 2.5 AU. It appears that within 2.5 AU, the local zodiacal intensity significantly reduces the performance of the interferometer and renders it no longer cost-effective. While separated spacecraft interferometers can produce more images, they are not as cost-effective as structurally connected interferometers. These



results point to total collector aperture area as the single most important parameter in developing a cost-effective TPF architecture.

In the second case study presented in Chapter 5, the Air Force Research Laboratory TechSat 21 military space-based radar mission, MMDOSA:

1. Found architectures that provide higher levels of performance (i.e. target probability of detection) for lower lifecycle costs than any of the proposed TechSat 21 point designs.
2. Identified a Pareto optimal set of system architectures that simultaneously minimize lifecycle cost, maximize target probability of detection, and minimize theater revisit time.
3. Determined that the cost-effectiveness of the GMTI mission can be improved by raising the constellation orbit from the current 800 km altitude to an altitude of 1300 km - 1500 km.

Increasing the constellation altitude reduces the total number of clusters required to meet the 15 minute theater maximum revisit time requirement, reducing the total number of satellites in the system. This in turn dramatically reduces lifecycle satellite procurement, launch, and operations costs. These cost savings more than compensate for the increase in cost that results from the increase in the number of satellites per cluster and the aperture diameter needed to project the power-aperture required to achieve the necessary target signal-to-noise ratio to overcome the increased signal path loss that occurs due to the increased range to target at higher altitudes. Future research should investigate the implications of moving the TechSat 21 constellation into the radiation belts.

In the third case study presented in Chapter 6, the commercial broadband satellite communications mission, MMDOSA:

1. Found architectures that provide higher levels of performance (i.e. greater system capacity) for lower lifecycle costs than any of the proposed industry designs.
2. Identified a Pareto optimal set of system architectures that simultaneously minimizes lifecycle cost, maximizes system capacity, maximizes the average number of satellites in view over the market served, and maximizes global availability.

3. Uncovered a direct correlation between constellation altitude, financial/program risk, and potential financial reward.

Interestingly, there was a smooth transition from GEO architectures at the low-cost, low-performance end of the Pareto optimal set through MEO architectures to LEO architectures at the high-cost, high-performance end of the Pareto optimal set. These results indicate that geosynchronous architectures provide a low-risk solution to the broadband satellite communications conceptual design problem, but with a limited return on investment. Conversely, low Earth orbit constellations represent an alternative high-risk solution to the same conceptual design problem. LEO constellations require a much greater up front investment, but provide the capacity for a greater return on investment as well. The actual return on investment of a system with such a large capacity will depend upon the size of the broadband market and the ability of satellite services to penetrate this market – both of which are highly uncertain variables in an uncertain marketplace. The most successful satellite broadband systems will be those that properly match their system capacity to the market demand. In each of the five business case scenarios explored, the most cost-effective – assuming the market absorbs the full system capacity – broadband satellite architectures utilize low power, large aperture satellites located in low Earth, medium to high inclination orbits. This third case study illustrates how the MMDOSA methodology can serve as a powerful, versatile tool for trade space exploration, concept development, and rapid analysis of alternative commercial space system business plans under uncertain market conditions.

## 7.2 Contributions

This thesis develops and validates a novel multiobjective, multidisciplinary approach to the conceptual design of distributed satellite systems. The following specific thesis contributions can be identified:

1. Developed a methodology for mathematically formulating and solving distributed satellite system conceptual design problems as nonlinear multiobjective optimization problems.

2. Determined that the heuristic simulated annealing technique finds the best (as defined by the metric(s) of interest) system architectures with greater consistency than Taguchi, gradient, and univariate techniques when searching a nonconvex distributed satellite system trade space.
3. Created two new multiobjective variants of the core single objective simulated annealing algorithm – the multiobjective single solution simulated annealing algorithm and the multiobjective multiple solution simulated annealing algorithm.
4. Created a method for computing the simulated annealing  $\Delta$ -parameter, originally developed and intended for single objective optimization problems, for multiobjective optimization problems.
5. Gathered empirical evidence that the 2-DOF variant of the simulated annealing algorithm is the most effective at both single objective and multiobjective searches of a distributed satellite system trade space.
6. Developed an integer programming approach to model and solve the distributed satellite system launch vehicle selection problem as an optimization problem.
7. For the NASA Terrestrial Planet Finder mission, identified specific architectures that provide higher levels of performance for lower lifecycle costs than any of the proposed industry designs, determined that structurally connected architectures are more cost-effective than separated spacecraft architectures for the planet detection and atmosphere characterization mission, and identified the performance limit of structurally connected architectures, beyond which separated spacecraft architectures represent the only feasible solution.
8. For the AFRL TechSat 21 mission, identified specific architectures that provide higher levels of performance for lower lifecycle costs than any of the proposed point designs, determined that the cost-effectiveness of the GMTI mission can be improved by raising the constellation orbit from the current 800 km altitude to an altitude of 1300 km - 1500 km, and identified a Pareto optimal set of architectures that simultaneously minimize lifecycle cost, maximize target probability of detection, and minimize theater revisit time.
9. For the commercial broadband satellite communications mission, identified specific architectures that provide higher levels of performance for lower lifecycle costs than any of the proposed industry designs; identified a direct correlation between constellation altitude, financial/program risk, and potential financial reward; and identified a Pareto optimal set of architectures that simultaneously minimizes lifecycle cost, maximizes system capacity, maximizes the average number of satellites in view over the market served, and maximizes global availability.

### 7.3 Recommendations for Future Work

Specific recommendations to further develop and extend this work include:

- Extension of the multiobjective optimization scope from optimizing an architecture with respect to multiple goals within the same mission to optimizing an architecture with respect to multiple goals for multiple missions (ex. optimizing the TechSat 21 system not only for the GMTI mission, but also for the geolocation, SAR, etc. missions.).
- Formally investigate and compare the performance of other heuristic algorithms – specifically genetic algorithms and the Tabu search algorithm – with the simulated annealing algorithm for DSS conceptual design.
- Development of reusable GINA models, modules, and databases that save the time and cost associated with creating these entities from scratch for each new mission study.
- Integration of the MMDOSA methodology with COTS aerospace analysis and mathematical optimization software, such as IDEAS and iSIGHT, as well as software, such as Oculus, that enables collaboration between a geographically dispersed conceptual design team.
- Incorporation of risk and uncertainty analysis tools to support decision makers analyzing MMDOSA results.
- Incorporation of Multi-Attribute Utility Analysis Theory to distinguish between architectures in higher-order Pareto optimal sets.
- Extension of the conceptual design trade space scope from structurally connected and separated space architectures to also include tethered and electromagnetic architectures for the NASA Terrestrial Planet Finder mission.
- Replacement of the current deconvolution radar signal processing model with other radar models, such as STAP and DPCA, to determine what effect the radar signal processing algorithm has on the selection of the optimal architectures for the AFRL TechSat 21 mission.
- Extension of the trade space scope to include elliptical orbits and hybrid constellations for the commercial broadband satellite communications mission.

# REFERENCES

- [Aarts & Lenstra, 1997] Aarts, Emile, and Lenstra, Jan, K., *Local Search in Combinatorial Optimization*, John Wiley & Sons, New York, NY, 1997.
- [Ahuja et al, 1993] Ahuja, Ravindra, K., Magnanti, Thomas, L., and Orlin, James, B., *Network Flows - Theory, Algorithms, and Applications*, Prentics Hall, Upper Saddle River, NJ, 1993.
- [Analytical Graphics, 2002] Analytical Graphics, *The Satellite Tool Kit (STK) Software Suite*, <http://www.stk.com/>, Analytical Graphics Inc., Malvern, PA, 2002.
- [ASQC, 1983] *Glossary and Tables for Statistical Quality Control*, 2nd Edition, American Society for Quality Control, Milwaukee, WI, 1983, pp. 85-86.
- [Barnett, 2001] Barnett, Arnold, *Chapter 3: Statistical Sampling*, 15.071 Course Notes, 2001.
- [Bearden & Law, 1996] Bearden, D.A., and Law, G.W., "An Integrated Approach to Small Concurrent Conceptual Design and Cost Estimation of Small Planetary Missions, *Proceedings of the 2nd IAA International Conference on Low-Cost Planetary Missions*, April 1996.
- [Bearden, 1998] Bearden, David, A., *A Methodology for Spacecraft Technology Insertion Analysis Balancing Benefit, Cost, and Risk*, Ph.D. Dissertation Proposal, Department of Aerospace Engineering, University of Southern California, August 1998.
- [Bearden, 1999] Bearden, David, A., *A Methodology for Spacecraft Technology Insertion Analysis Balancing Benefit, Cost, and Risk*, Doctoral Thesis, Department of Aerospace Engineering, University of Southern California, Los Angeles, CA May 1999.
- [Bearden, 2000] A "Complexity-based Risk Assessment Of Low-cost Planetary Missions: When Is A Mission Too Fast And Too Cheap?" *Proceedings of the Fourth Iaa International Conference On Low-cost Planetary Missions*, , John's Hopkins University/Applied Physics Laboratory, Laurel, MD, May 2000.
- [Beichman, 1998] Beichman, C.A., "The Terrestrial Planet Finder: The Search for Life-Bearing Planets Around Other Stars. *Proceedings of the SPIE Conference on Astronomical Interferometry*, SPIE Col. 3350, Part Two, Kona, Hawaii, March 1998, pp. 719-723.
- [Beichman et al, 1999] Beichman, C.A., Woolf, N.J., and Lindensmith, C.A., "The Ter-

- restrial Planet Finder (TPF): A NASA Origins Program to Search for Habitable Planets," JPL Publication 99-3, May 1999, pp. 1-11, 49-55, 87-89.
- [Bertsimas & Tsitsiklis, 1997] Bertsimas, Dimitris, and Tsitsiklos, John, N., *Introduction to Linear Optimization*, Athena Scientific, Belmont, MA, 1997.
- [Bertsimas & Freund, 2000] Bertsimas, Dimitris, and Freund, Robert, M., *Data, Models, and Decisions: The Fundamentals of Management Science*, South-Western College Publishing, Cincinnati, OH, 2000.
- [Boden & Larson, 1996] Boden, D.G., and Larson, W.J., *Cost-Effective Space Mission Operations*, McGraw Hill Inc., New York, NY, 1996.
- [Boeing, 1999] The Boeing Company, *Application for Authority to Launch and Operate a Non-Geostationary Medium Earth Orbit Satellite System in the Fixed Satellite Service*, Seattle, WA, FCC Filing, January 1999
- [Boppe, 1997] Boppe, C.W., *System Engineering Elements Addressing Complexity*, MIT Course 16.870, Aerospace Product Design Course Notes, Cambridge, MA, 1997.
- [Boppe, 1997] Boppe, C.W., *Physical Interfaces and Schematic Block Diagrams*, MIT Course 16.870, Aerospace Product Design Course Notes, Cambridge, MA, 1997.
- [Bos, 1996] Bos, D., "Multidisciplinary Design Optimization of a Supersonic Transport Aircraft using a Hybrid Genetic/Gradient-Based Algorithm" AIAA-96-4055-CP, 1996.
- [Bracewell & Macphie, 1979] Bracewell, R.N., and Macphie, R.H., "Searching for Non Solar Planets," *Icarus*, 1979, pp. 136.
- [Braun et al, 1995] Braun, R.D., Powell, R.W., Lepsch, R.A., Stanley, D.O., and Kroo, I.M., "Comparison of Two Multidisciplinary Optimization Strategies for Launch Vehicle Design," *Journal of Spacecraft and Rockets*, Vol. 32, No. 3, 1995, pp. 404-410.
- [Braun et al, 1996] Braun, R., Gage, P., Kroo, I., and Sobieski, I., "Implementation and Performance Issues in Collaborative Optimization," *Proceedings of the 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA-96-4017-CP, Bellevue, WA, Sept. 1996, pp. 295-305.
- [Braun et al, 1996] Braun, R.D., Moore, A.A., and Kroo, I.M., "Use of the Collaborative Optimization Architecture for Launch Vehicle Design," *Proceedings of the 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA-96-4018-CP, Bellevue, WA, Sept. 1996, pp. 306-318.
- [Briggs, 1995] Briggs, C.W., "Reengineering the JPL Spacecraft Design Process," *Pro-*

---

*ceedings of the IFORS 4th Specialized Conference in Operations Research in Engineering Design*, October 1995.

- [Brooks et al, 1998] Brooks, R.R., Iyenger, S.S., and Rai, S., "Comparison of Genetic Algorithms and Simulated Annealing for Cost Minimization in a Multisensor System," *SPIE Journal of Optical Engineering*, Vol. 37(2), February 1998, pp. 505-516.
- [Browning, 1998] Browning, T.R., *Modeling and Analyzing Cost, Schedule, and Performance in Complex System Product Development*, Doctoral Thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 1998.
- [Buckingham et al, 1995] Buckingham, R., Payne, W.F., and Sultan, N., "Two Methods to Predict and Constrain Small Satellite and Payload Cost," *Proceedings of the 9th Annual AIAA/USU Conference on Small Satellites*, Logan, UT, September 1995.
- [Cantafio, 1989] Cantafio, L.J., ed., *Space-Based Radar Handbook*, Chapter 11, Artech House Inc., Norwood, MA, 1989.
- [Casani, 1994] Casani, Kane, *Reengineering the JPL Project Design Process*, August 1st, 1994.
- [Curtis et al, 1999] Curtis, A., de Weck, O., Frazzoli, E., Girerd, A., Hacker, T., Jilla, C., Kong, E., Makins, B., and Pak, S., *ASTRO – Architecting the Search for Terrestrial Planets and Related Origins*, Course 16.89 Space Systems Engineering Final Report, SERC # 6-99, MIT Space Systems Laboratory, Cambridge, MA, 1999.
- [Das & Cobb, 1998] Das, A., and Cobb, R., "TechSat 21 - Space Mission Using Collaborating Constellations of Satellites," *Proceedings of the 12th AIAA/USU Conference on Small Satellites*, Paper SSC98-VI-1, Utah State University, Logan, UT, Sept. 1998.
- [de Neufville, 1990] de Neufville, Richard, *Applied Systems Analysis: Engineering Planning and Technology Management*, McGraw-Hill, New York, NY, 1990.
- [Engler et al, 1998] Engler, V., Coors, D., and Jacobs, D., "Optimization of a Space Transportation System Including Design Sensitivities," *Journal of Spacecraft and Rockets*, Vol. 35, No. 6., 1998, pp. 785-791.
- [Enright et al, 1999] Enright, J., Sedwick, R., and Miller, D., "Information Architecture Analysis and Optimization for Space-Based Distributed Radar," *Proceedings of the AIAA Space Technology Conference*, AIAA-99-4551, Albuquerque, NM, September 1999.

- [Francis et al, 1997] Francis, B., et al., "Lockheed Martin TPF Final Review," *Lockheed Martin Missiles & Space TPF Conceptual Design Report*, 1997.
- [Fukunaga et al, 1997] Fukunaga, A.S., Chien, S., Mutz, D., Sherwood, R.L., and Stechert, A.D., "Automating the Process of Optimization in Spacecraft Design," *Proceedings of the 1997 IEEE Aerospace Conference*, 1997, pp. 411-427.
- [Gelb, 1974] Gelb, A., ed., *Applied Optimal Estimation*, The MIT Press, Cambridge, MA, 1974, pp.42-45.
- [Goldberg, 1989] Goldberg, David, E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, MA, 1989, pp. 197-201.
- [Greenberg, 1992] Greenberg, J., "A Computer-Aided Approach for Earth Observation Spacecraft Design and Technology Evaluation," *Proceedings of the 43rd International Astronautical Congress*, 1992.
- [Hacker & Sedwick, 1999] Hacker, T.L., and Sedwick, R.J., "Space-Based GMTI Radar System Using Separated Spacecraft Interferometry," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA-99-4634, Portland, OR, Aug. 1999.
- [Hacker et al, 2000] Hacker, T.L., Sedwick, R.J., and Miller, D.W., *Performance of a Space-Based GMTI Radar System Using Separated Spacecraft Interferometry*, SERC Report # 2-2000, MIT Space Systems Laboratory, Cambridge, MA, May 2000.
- [Hale, 1994] Hale, F.J., *Introduction to Space Flight*, Prentice Hall, Englewood Cliffs, NJ, 1994, pp. 42-50.
- [Hansen & Vanderplaats, 1990] Hansen, S.R., and Vanderplaats, G.N., Approximation Method for Configuration Optimization of Trusses, *AIAA Journal*, Vol. 28, No. 1, 1990, pp. 161-168.
- [Hickman, 1998] Hickman, R.A., "Integrated Modeling for Launch System Architecture Optimization," *Proceedings of the 7th AIAA Multidisciplinary Design Conference*, St. Louis, MO, 1998.
- [Hopkins, 1990] Hopkins, H., *Investigation of Direct and Indirect Optimization Algorithms for Aerospace Structures*, M.S. Thesis, Air Force Institute of Technology, AIAA Technical Library, 1990.
- [Hsu & Adeli, 1991] Hsu, H.L., and Adeli, H., "A Microtasking Algorithm for Optimization of Structures," *International Journal of Supercomputer Applications*, Vol. 5, Summer, 1991, pp. 81-90.



- 
- [Huddleston, 1999] Huddleston, D.W., *Aerodynamic Design Optimization Using Computational Fluid Dynamics*, Ph.D. Thesis, University of Tennessee, Knoxville, TN 1989.
- [Hughes Communications, 1999] Hughes Communications Inc., *Application of Hughes Communications, Inc. for Authority to Launch and Operate HughesNET, a Global Ku band Broadband Satellite System*, Washigton D.C., FCC Filing, January 1999.
- [Hughes Communications, 1999] Hughes Communications Inc., *Application of Hughes Communications, Inc. for Authority to Launch and Operate HughesLINK, a Global Ku band Broadband Satellite System*, Washigton D.C., FCC Filing, January 1999.
- [INCOSE, 1998] INCOSE, *Systems Engineering Handbook*, San Francisco Bay Area Chapter International Council on Systems Engineering, Techical Report, January 1998.
- [Isakowitz, 1995] Isakowitz, S.J., *International Reference Guide to Space Launch Systems*, 2nd ed., American Institute of Aeronautics and Astronautics, Washington D.C., 1995.
- [Jilla & Miller, 1997] Jilla, C.D., and Miller, D.W., "A Reliability Model for the Design and Optimization of Separated Spacecraft Interferometer Arrays," *Proceedings of the 11th AIAA/USU Conference on Small Satellites*, Paper SSC97-XI-2, Utah State University, Logan, UT, Sept. 1997.
- [Jilla et al, 2000] Jilla, C.D., Miller, D.W., and Sedwick, R.J., "Application of Multidisciplinary Design Optimization Techniques to Distributed Satellite Systems," *Journal of Spacecraft and Rockets*, Vol. 37, No. 4, 2000, pp. 481-490.
- [Jilla & Miller, 2000] Jilla, C.D., and Miller, D.W., "Assessing the Performance of a Heuristic Simulated Annealing Algorithm for the Design of Distributed Satellite Systems," *Proceedings of the 51st International Astronautical Congress*, IAF-00-U.2.02, Rio de Janeiro, Brazil, October 2000.
- [Jilla & Miller, 2001] Jilla, C.D., and Miller, D.W., "Assessing the Performance of a Heuristic Simulated Annealing Algorithm for the Design of Distributed Satellite Systems," *Acta Astronautica*, Vol. 48, No. 5-12, 2001, pp. 529-543.
- [Jonas et al, 1999] Jonas, F., Staggs, J., and Johnson, E., "The Transmit/Receive Antenna Module II (TRAM II)," *Proceedings of the AIAA 1999 Space Technology Conference*, AIAA-99-4562, Albuquerque, NM, September 1999.
- [Kashitani, 2002] Kashitani, Tatsuki, "Development and Application of an Analysis Methodology for Satellite Broadband Network Architectures," *Proceedings of*

*the 20th AIAA International Communication Satellite Systems Conference and Exhibit*, AIAA-2002-2019, Montreal, Canada, May 2002.

[Kashitani, 2002] Kashitani, Tatsuki, *Development and Application of an Analysis Methodology for Satellite Broadband Network Architectures*, Masters of Engineering Thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 1998.

[KB Sciences, 1994] KB Sciences, *Space Mission Analysis and Design Software User's Guide*, KB Sciences, Colorado Springs, CO, 1994.

[Kesselman, 1998] Kesselmann, M., "Optimization and Evaluation of Single-Staged Transportation Systems Example for the Solution of Highly Complex Technical Problems," *Proceedings of the AIAA Multidisciplinary Design Conference*, St. Louis, MO, 1998.

[Kirkpatrick et al, 1983] Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P., "Optimization by Simulated Annealing," *Science*, Volume 220, Number 4598, 13 May 1983, pp. 671-680.

[Klemm, 1998] Klemm, R., *Space-Time Adaptive Processing: Principles and Applications*, Peter Peregrinus Limited, 1998.

[Kockler, F.R., 1990] Kockler, F.R., *Systems Engineering Management Guide*, Government Printing Office, 1990.

[Kroo et al, 1994] Kroo, I., Altus, S., Braun, R., Gage, P., and Sobieski, I., "Multidisciplinary Optimization Methods for Aircraft Preliminary Design", *Proceedings of the 5th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, September, 1994.

[Larson & Wertz, 1992] Larson, W.J., and Wertz, J.R., *Space Mission Analysis and Design*, 2nd Edition, Microcosm Inc., Torrance, CA, 1992.

[Leech, 1956] Leech, J., "On the Representation of 1,2,...n by Differences," *Journal of the London Mathematical Society*, Vol. 31, 1956.

[Logothetis and Wynn, 1989] Logothetis, N., and Wynn, H.P., *Quality Through Design: Experimental Design, Off-Line Quality Control, and Taguchi's Contributions*, Clarendon Press, Oxford, Great Britain, 1989, pp. 1-20.

[Marais & Sedwick, 2001] Marais, K., Sedwick, R.J., *The Development and Analysis of Scanned Pattern Interferometric Radar*, Masters Thesis, Massachusetts Institute of Technology, Cambridge, MA, 2001.

[Martin & Stallard, 1999] Martin, M., and Stallard, M.J., "Distributed Satellite Missions

- and Technologies - The TechSat 21 Program," *Proceedings of the AIAA 1999 Space Technology Conference*, AIAA-99-4479, Albuquerque, NM, September 1999.
- [*Martin, 2000*] Martin, Maurice, "TechSat21 Program Overview," *TechSat21: Sparse Aperture Sensing Using Reconfigurable Microsatellite Formations*, Presentation to Independent Review Board, Albuquerque, NM, May 2000.
- [*Mason et al, 1998*] Mason, W.J., Coverstone-Carroll, V., and Hartman, J.W., "Optimal Earth Orbiting Satellite Constellations Via a Pareto Genetic Algorithm," *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Boston, MA, 1998, pp. 169-177.
- [*Massardo & Satta, 1989*] Massardo, A., Satta, A., "Axial Flow Compressor Design Optimization," *Proceedings of the ASME Gas Turbine and Aeroengine Congress and Exposition*, ASME Paper # 89-GT-201, 1989.
- [*Mathaisel, 1997*] Mathaisel, D.F.X., "Decision Support for Airline Schedule Planning," *Journal of Combinatorial Optimization*, 1997, pp. 251-275.
- [*Mathworks, 2002*] Mathworks, *The Mathworks Home: Increasing the Scope and Productivity of Engineering and Science*, <http://www.mathworks.com/>, The Mathworks Inc., Natick, MA, 2002.
- [*Matossian, 1995*] Matossian, M., "Earth Observing System Mission Design: Constrained Optimization of the EOS Constellation Configuration Design," *Proceedings of the 46th International Astronautical Congress*, IAF-95-A.6.02, 1995.
- [*Matossian, 1995*] Matossian, M., "Design for Success: Optimizing the Earth Observing System for Performance and Cost with Managed Risk," *Proceedings of the 46th International Astronautical Congress*, IAF-95-Q.4.05, 1995.
- [*Matossian, 1996*] Matossian, M., "Distributed Task Constellation Design Optimization: An Operations Research Approach to Earth Observing System Configuration Design," *Proceedings of the 47th International Astronautical Congress*, IAF-96-U.4.07, 1996.
- [*Matossian, 1996*] Matossian, M.G., "Earth Observing System Constellation Design Through Mixed Integer Programming," *Space Technology*, Vol. 16, No. 5/6, 1996, pp. 233-243.
- [*Matthews, 1998*] Matthews, C. *Case Studies in Engineering Design*, Arnold Publishers, London, 1998.
- [*Mennesson & Mariotti, 1997*] Mennesson, B., and Mariotti, J.M., "Array Configurations for a Space Infrared Nulling Interferometer Dedicated to the Search for Earthlike

- Extrasolar Planets," *Icarus* 128, 1997, pp. 202-212.
- [Merriam-Webster, 1998] Merriam-Webster Inc., *Merriam-Webster's Collegiate Dictionary*, 10th edition, International Thomson Publishing, Springfield, MA, 1998.
- [Microcom Systems, 2002] Microcom Systems, "Broadband Satellites," White Paper, Microcom Systems Ltd, PO Box 21, Haverhill, Suffolk CB9 0NZ, England, <http://www.satelliteonthenet.co.uk/white/bband.html>, 2002.
- [Miller et al, 2000] Miller, D.W., Curtis, A., De Weck, O., Frazzoli, E., Girerd, A., Hacker, T., Jilla, C.D., Kong, E.M., Makins, B., and Pak, S., "Architecting the Search for Terrestrial Planets and Related Origins (ASTRO)," *Proceedings of the SPIE International Symposium on Astronomical Telescopes and Instrumentation 2000*, SPIE 4013-74, Munich, Germany, March 2000.
- [Mosher, 1996] Mosher, T., "Applicability of Selected Multidisciplinary Design Optimization Methods to Conceptual Spacecraft Design," *Proceedings of the 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA-96-4052, Bellevue, WA, Sept. 1996, pp.664-671.
- [Mosher, 1998] Mosher, T., "Spacecraft Design Using a Genetic Algorithm Optimization Approach," *Proceedings of the 1998 IEEE Aerospace Conference*, Volume 3 of 5, Aspen, CO, 1998, pp. 123-134.
- [Mosher, 2000] Mosher, Todd, J., *Improving Spacecraft Design Using a Multidisciplinary Design Optimization Methodology*, Doctoral Thesis, Department of Aerospace Engineering Sciences, University of Colorado, Boulder, CO, 2000.
- [Motorola, 1997] Motorola Global Communications Inc., *Application for Authority to Construct, Launch, and Operate the Celestri Multimedia LEO System, A Global Network of Non-Geostationary Communications Satellites Providing Broadband Services in the Ka-Band*, FCC Filing, June 1997.
- [Munson & Jilla, 2000] Munson, J.E., and Jilla, C.D., "An Optimization Approach to the Launch Vehicle Selection Process for Satellite Constellations," *Proceedings of the 18th AIAA International Communications Satellite Systems Conference*, AIAA-2000-1122, Oakland, CA, April 2000.
- [Murray et al, 1997] Murray, D.J., Coe, D.J., and White, R.G., "Experimental MTI With Spaceborne Geometries," *Proceedings of the IEE Colloquium on Radar Interferometry*, Ref. No. 1997/153, April 1997.
- [Noecker et al, 1997] Noecker, C., et al., "Ball TPF Study," *Ball TPF Conceptual Design Report*, 1997.
- [Pareto, 1906] Pareto, Vilfredo, *Manuale di Politica*, Societa Editrice Libreria, Milano,

- 
- Italy, 1906. Translated into English by A.S. Schwier as *Manual of Political Economy*, Macmillan, New York, NY, 1971.
- [Pioneer Consulting, 1998] Pioneer Consulting, *Global Broadband Access Markets: xDSL and Cable Modems Threatened by Broadband Satellite, Wireless and All-Optical Solutions*, Pioneer Consulting LLC, Cambridge, MA, October 1998.
- [Principia Cybernetica, 2002] Principia Cybernetica Web, *Web Dictionary of Cybernetics and Systems*, <http://pespmc1.vub.ac.be//ASC/OPTIMIZATIO.html>, Optimization, 2002.
- [Rabideau & Kogon, 1999] Rabideau, D.J., and Kogon, S.M., "A Signal Processing Architecture for Space-Based GMTI Radar," *Proceedings of the IEEE Radar Conference*, April 1999.
- [Riddle, 1998] Riddle, E., "Use of Optimization Methods in Small Satellite Systems Analysis" *Proceedings of the 12th AIAA/USU Conference on Small Satellites*, Utah State University, Logan, UT, Paper SSC98-X-1, Sept. 1998.
- [Rolle, 1999] Rolle, Amy, Iridium Operations Center Engineer, Presentation at MIT to the 16.899 Class, September 24th, 1999.
- [Ross, 1996] Ross, P.J., *Taguchi Techniques for Quality Engineering: Loss Function, Orthogonal Experiments, Parameter and Tolerance Design*. 2nd ed., McGraw-Hill, New York, NY, 1996, pp. 208-212.
- [Rowell et al, 1996] Rowell, L.F., Braun, R.D., Olds, J.R., and Unal, R., "Recent Experiences in Multidisciplinary Conceptual Design Optimization for Launch Vehicles," *Proceedings of the 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA-96-4050-CP, Bellevue, WA, Sept. 1996, pp. 642-653.
- [Saleh & Hastings, 2000] Saleh, J.H., and Hastings, D.E., "On Flexibility In Design: Analyzing Flexibility of Space Systems," *Proceedings of the AIAA Space 2000 Conference and Exposition*, Long Beach, CA, Sept. 2000.
- [Saleh, 2002] Saleh, Joseph, H., *Weaving Time Into System Architecture: New Perspectives on Flexibility, Spacecraft Design Lifetime, and On-Orbit Servicing*, Doctoral Thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 2002.
- [Sedwick et al, 1998] Sedwick, R.J., Kong, M.C., and Miller, D.W., "Exploiting Orbital Dynamics and Micropropulsion for Aperture Synthesis Using Distributed Satellite Systems: Applications to TechSat 21," *Proceedings of the AIAA Defense & Civil Space Programs Conference*, AIAA-98-5289, Huntsville, AL, October 1998.

- [Sedwick et al, 1999] Sedwick, R.J., Hacker, T.L., and Miller, D.W., "Optimum Aperture Placement for a Space-Based Radar System Using Separated Spacecraft Interferometry," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA-99-4271, Portland, OR, August 1999.
- [Sedwick et al, 1999] Sedwick, R.J., Miller, D.W., and Kong, E.M., "Mitigation of Differential Perturbations in Clusters of Formation Flying Satellites," *The Journal of the Astronautical Sciences*, Vol. 47, Nos. 3 and 4, 1999, pp. 309-331.
- [Sercel et al, 1998] Sercel, J., Sepahban, S., and Wall, S., "ICE Heats Up Design Productivity," *Aerospace America*, July 1998, pp. 20-22.
- [Senin et al, 1999] Senin, N., Wallace, D.R., and Borland, "Object-Based Design Modeling and Optimization with Genetic Algorithms," *Proceedings of the GECCO99 - Real World Applications Conference*, 1999.
- [Shaw, 1998] Shaw, Graeme, B., *The Generalized Information Network Analysis Methodology for Distributed Satellite Systems*, Doctoral Thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 1998.
- [Shaw, 1998] Shaw, G.B., *The Generalized Information Network Analysis (GINA) Methodology for Distributed Satellite Systems*, Doctoral Dissertation Defense Presentation, Massachusetts Institute of Technology, Cambridge, MA, 1998.
- [Shaw et al, 2000] Shaw, G.B., Miller, D.W. and Hastings, D.E., "Generalized Characteristics of Satellite Systems," *Journal of Spacecraft and Rockets*, Vol. 37, No. 6, 2000, pp. 801-811.
- [Shaw et al, 2001] Shaw, G.B., Miller, D.W., and Hastings, D.E., "Development of the Quantitative Generalized Information Network Analysis (GINA) Methodology for Satellite Systems," *Journal of Spacecraft and Rockets*, Vol. 38, No. 2, 2001, pp. 257-269.
- [Shishko & Chamberlain, 1995] Shishko, R., and Chamberlain, R., *NASA Systems Engineering Handbook*, SP-6105, NASA, June 1995, pp. 3-7, 13-26, 67-90.
- [Silverstein, 1999] Silverstein, Sam, "ICO Struggles to Raise Capital: Iridium Declares Bankruptcy as Competitor Seeks Funds," *Space News*, Vol. 10, No. 32, August 23, 1999.
- [Skinner, 1992] Skinner, D.L., "Spacecraft Design Applications of QUICK," *Proceedings of the AIAA Aerospace Design Conference*, February 1992.
- [Skolnik, 1980] Skolnik, M.I., *Introduction to Radar Systems*, Chapter 18, McGraw-Hill, New York, 1980.

- 
- [SkyBridge, 1997] SkyBridge LLC, *Application of SkyBridge L.L.C. for Authority to Launch and Operate the SkyBridge System, a Global Network of Low Earth Orbit Communications Satellites Providing Broadband Services in the Fixed Satellite Service*, Washington D.C., FCC Filing, February 1997.
- [SkyBridge, 1999] SkyBridge LLC, *Amendment to Application of SkyBridge L.L.C. for Authority to Launch and Operate the SkyBridge System*, Washington D.C., FCC Filing Amendment, January 1999.
- [Sobieczky, 1990] Sobieczky, H., *Progress in Inverse Design and Optimization in Aerodynamics*, Institute for Theoretical Fluid Mechanics, Germany, AIAA Technical Library, 1990.
- [Sobieszcanski-Sobieski, 1990] Sobieszcanski-Sobieski, J., "Sensitivity Analysis and Multidisciplinary Optimization for Aircraft Design: Recent Advances and Results", *Journal of Aircraft*, Volume 27, No. 12, December, 1990.
- [Sobieszcanski-Sobieski, 1993] Sobieszcanski-Sobieski, J., "Multidisciplinary Design Optimization: An Emerging New Engineering Discipline," *Proceedings of the World Congress on Optimal Design of Structural Systems*, Rio de Janeiro, Brazil, August 1993.
- [Stallard, 1999] Stallard, M., *TechSat 21 Satellite Technology and Conceptual Design*, Presentation to Integrated Space Experiments Review Board, Albuquerque, NM, February 1999.
- [Stettner, 1999] Stettner, L., "Risk Sensitive Portfolio Optimization," *Mathematical Methods of Operations Research*, December 1999, pp. 463-474.
- [Stimson, 1998] Stimson, G.W., *Introduction to Airborne Radar*, 2nd ed., SciTechPublishing, Mendham, NJ, 1998.
- [Subramanian et al, 1999] Subramanian, J., Stidham, S., and Lautenbacher, C.J., "Airline Yield Management with Overbooking, Cancellations, and No-Shows," *Transportation Science*," May 1999, pp. 147-167.
- [USAF, 1996] United States Air Force, *New World Vistas – Air and Space Power for the 21st Century*, 1986.
- [Walker, 1971] Walker, J.G., "Some Circular Orbit Patterns Providing Continuous Whole Earth Coverage," *Journal of the British Interplanetary Society*, Vol. 24, July, 1971, pp. 369-384.
- [Walton & Hastings, 2001] Walton, M., and Hastings, D., "Quantifying Embedded Uncertainty of Space Systems Architectures in Conceptual Design," *Proceedings of the AIAA Space 2001 Conference*, AIAA-2001-4573, Albuquerque, NM,

August 2001.

[Wehner et al, 1997] Wehner, M., Johnson, L., Kroening, K., and Moses, S., "TRW Terrestrial Planet Finder Study," *TRW TPF Conceptual Design Report*, 1997.

[Weigel & Hastings, 2001] Weigel, A.L., and Hastings, D.E., "Interaction of Policy Choices and Technical Requirements for a Space Transportation Infrastructure," *Proceedings of the 5th International Conference on Technology, Policy, and Innovation*, The Hague, Netherlands, June 2001.

[Wertz & Larson, 1996] Wertz, J.R., and Larson, W.J., *Reducing Space Mission Cost*, Microcosm, Torrance, CA, 1996.

[Wilson, 2000] Wilson, J.R., "Satellite Clusters Reshape Space Plans," *Aerospace America*, November 2000, pp. 27-30.

[Wu and Lake, 1996] Wu, K.C., and Lake, M.S., "Multicriterion Preliminary Design of a Tetrahedral Truss Platform," *Journal of Spacecraft and Rockets*, Vol. 33, No. 3, May-June 1996, pp. 410-415.



# Appendix A

## BENEFITS OF DISTRIBUTED SATELLITE SYSTEMS

### A.1 Overview

This Appendix lists and discusses some of the potential benefits of distributed architectures for space missions, and classifies these benefits in accordance with the GINA metrics. For a complete discussion on the theory behind and the classification scheme for distributed satellite systems, please see the following two references:

- Shaw, Graeme, B., *The Generalized Information Network Analysis Methodology for Distributed Satellite Systems*, Doctoral Thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 1998.
- Shaw, G.B., Miller, D.W. and Hastings, D.E., "Generalized Characteristics of Satellite Systems," *Journal of Spacecraft and Rockets*, Vol. 37, No. 6, 2000, pp. 801-811.

### A.2 Potential Benefits of Distribution

#### A.2.1 List of the Potential Benefits of Distribution

1. Decentralization of Resources
2. Smaller, Simpler Satellites
3. Enables "Mass Production" of Satellites
4. Promotes Modular Designs
5. Spatial Distribution of Payloads
6. Reduced Range to Target

7. Enables Separate, Sparse Apertures
8. Multiple Viewing Angles
9. Reduced Computational Workload
10. More Satellites
11. Task Division
12. Increases SNR
13. Clutter Rejection
14. Redundancy
15. Path Diversity
16. Allows for Ramped Deployment and Upgrades
17. Minimizes Required Failure Compensation Costs
18. Multi-Mission Capability

### **A.2.2 Explanation of the Potential Benefits of Distribution**

#### **1. Decentralization of Resources**

The "resources" of a satellite system are the payloads that provide the desired information to the customer, whether that information be scientific data for a civilian agency, imagery for a defense agency, multimedia for a commercial organization, or navigation data for any of the above three entities. Decentralization refers to the fact that in a distributed satellite system, these payloads are dispersed among several spacecraft. For example, a suite of Earth observation instruments that would conventionally fly on a single polar orbiting weather satellite in LEO might be distributed between three separate, smaller satellite buses in the same orbital plane. In the original configuration, a failure of the spacecraft bus, such as might occur due to the malfunction of the attitude control computer, would render the entire suite of instruments (i.e. *all* of the resources) useless. In the DSS configuration, only the payloads on the bus that fails are lost, while the instruments on the two remaining spacecraft may still be used. Thus, the decentralization of resources may improve the *availability* of a system. Likewise, decentralization of assets in a military space system can improve the *survivability* of the system in times of conflict.

#### **2. Smaller, Simpler Satellites**

By distributing resources on multiple satellite platforms rather than placing all instruments on a single platform, smaller satellite buses may be used. This in turn results in a direct *savings in the manufacturing cost* of each satellite. Additionally, the production of smaller, simpler satellite buses is

faster than for larger satellite buses. Thus, the use of distributed architectures can also *reduce the time to the Initial Operating Capability (IOC)* of a space system.

### 3. "Mass Production" of Satellites

By employing the use of multiple identical satellite platforms, distributed satellite systems motivate the "mass production" of satellites. The resulting economies of scale promote *learning curve savings*, reducing the cost of each individual satellite. Additionally, mass production of identical units also simplifies the integration and testing process, *reducing the time to IOC* and potentially *improving the reliability* of the system.

### 4. Modular Designs

Distributed satellite systems promote modular designs. A modular satellite bus design is one which provides the capability to support different payloads. Different components of the bus may be incrementally upgraded or downgraded depending upon the characteristics of the given payload. Take the example of the three LEO weather satellites orbiting in the same plane. One satellite may require larger solar arrays and more batteries because its payload suite requires more power than the payload suites on the other two satellites, while another satellite might suffice with smaller reaction wheels because its payload is less massive. While such particular characteristics may vary between satellites, the core bus design remains the same in a modular design. Modular designs promote mass production of satellite buses, leading to the same *reductions in cost and time to IOC* as previously explained. Modular designs also allow for easy *upgradability*, as newer, more advanced payload and bus technologies may be integrated with the same modular bus by employing standardized interfaces.

### 5. Spatial Distribution

Spatial distribution involves the use of multiple satellites in different orbits. Using more satellites in a distributed system *reduces the revisit time* to a particular target on the Earth. Spatial distribution can also improve the *isolation, rate, and integrity* capabilities of a space system.

### 6. Reduced Range to Target

In certain cases, utilization of a distributed architecture can reduce the required range to target. Take the example of a space based-radar (SBR) system designed to identify and track military targets in theaters of conflict. A conventional design employs a few satellites in MEO or GEO to cover the globe. Since each satellite must be capable of satisfying the radar mission individually over its field of view, large diameter antennas and tremendous amounts of power are required to execute the mission. This results in the use of large, expensive satellites. A distributed architecture may employ the use of symbiotic clusters to fulfill the same mission. Since each target is now

illuminated by multiple spacecraft rather than a single spacecraft, smaller antennas and lower Earth orbits may be used. This in turn *reduces the total power-aperture* required of each satellite, which in turn *reduces the total cost* of the system. Reducing the range to target also improves the *isolation capability* of a system (higher resolution for optical payloads) and the *integrity* of the information being transmitted throughout the network.

### **7. Separated, Sparse Apertures**

Distributed satellite systems promote the use of separated, sparse apertures for surveillance missions. While the angular resolution of a conventional single aperture telescope is proportional to the diameter of the primary mirror of that telescope, the angular resolution of a sparse aperture is proportional to the maximum baseline between two elements (mirrors) of the sparse aperture. Thus, distributed satellite systems enable greater angular resolution than possible with single satellite designs, improving the *isolation capability* of the system.

### **8. Multiple Viewing Angles**

Spatially distributing resources over many satellites in a Distributed Satellite System also increases the number of viewing angles between the system and the target. For example, increasing the number of satellites in a telephony constellation decreases the probability that the user telephone on the ground will not be able to communicate with a satellite in the constellation via a direct link due to line-of-sight obstruction by buildings and other tall objects. Similarly, the more satellites there exist in a global navigation system, the more likely the GPS receiver on the ground will be able to identify four satellites with good viewing angles to enable an accurate positioning measurement. Thus, the multiple viewing angles promoted by Distributed Satellite Systems improve the *availability* and *rate* of space-based systems.

### **9. Reduced Computational Workload**

Distributed satellite systems distribute the resources of a space system across many platforms. One these resources are space-qualified computers. Due to time consuming radiation-hardening and other qualification procedures, the capability (speed and memory) of space-qualified computers is always several generations behind their ground-based counterparts. Certain missions however, such as the space-based radar (SBR) mission, require a tremendous amount of computational power to process the signal data and identify targets in real-time. No space-qualified computer is capable of executing such computations in real time, making it unlikely that a single SBR satellite could safely cover the theater of interest. A DSS approach to the SBR mission divides the computations among the many computers on different platforms. By processing different portions of the same signal information in parallel, such a complicated mission might just become feasible with a dis-

tributed satellite system architecture. Reducing the computational workload in this manner reduces computational *costs* while increasing the system *rate*.

#### **10. More Satellites**

Distributed satellite systems promote the use of many smaller satellites over a few larger satellites. Having more total satellites in a space system improves *survivability*, *revisit times*, and *coverage geometry*. If properly designed, more satellites can also improve the total capability of the system in terms of *isolation*, *rate*, *integrity*, and *availability*.

#### **11. Task Division**

In a collaborative DSS architecture, the responsibilities of each satellite decrease linearly with the number of satellites in the system. Each satellite can allocate more of its resources to each source, satisfying *higher rate* requirements. Increasing the number of satellites in the cluster yields linear increases in the achievable rate of information flow from each source. The limit is reached when each satellite is dedicated to a single user. The maximum rate for that user is the maximum supportable rate of the satellite.

#### **12. Increases SNR**

In a symbiotic DSS architecture, the SNR of an active or passive surveillance system increases proportionally to the number of satellite in the system. Higher SNR's in turn boost the *integrity* and *rate* of the system.

#### **13. Clutter Rejection**

Symbiotic distributed satellite systems can help to reduce the clutter in certain applications. Once again, take the example of an SBR system. As a direct result of the smaller beamwidths that are characteristic of symbiotic systems, the clutter rejection of the system is greatly improved compared to single satellites or collaborative systems. This in turn improves the measured SNR and increases the *integrity* and *rate* of the system.

#### **14. Redundancy**

Distributed satellite systems provide an inherent redundancy in space systems. Failure of a single satellite within a cluster or constellation does not prevent the system from accomplishing its mission, but may decrease the system capability. This holds true for all of the example missions previously discussed - weather surveillance, SBR, and telecommunications. Thus, the redundancy provided by distributed satellite systems improves the *reliability* and *availability* of space assets.

#### **15. Path Diversity**

The benefits of path diversity in distributed satellite systems parallel those provided by redundancy. According to the Generalized Information Network Analysis (GINA) theory, any satellite system may be transformed into

an information network where each satellite or satellite payload is a node in the network. Distributed satellite systems provide more nodes in the network through which information may flow than conventional single satellite deployments. In the case of a satellite failure, there are more alternative routes in the DSS for the information to arrive to the final user. In this manner, path diversity also improves the *reliability* and *availability* of a space system.

#### **16. Allows for Ramped Deployment and Upgrades**

Collaborative distributed systems offer the possibility of being able to ramp up the investment gradually to match the development of the market. Only those satellites needed to satisfy the early market are initially deployed. If and when additional demand develops, the constellation can be augmented. The cost of constructing and launching these additional satellites is incurred later in the system lifetime. Due to the time value of money, the delayed expenditure can result in significant *cost* savings.

#### **17. Minimization of Failure Compensation Costs**

The decentralization of resources, inherent redundancy, and path diversity in distributed satellite systems minimize failure compensation costs. Recall the polar orbiting weather satellite system in LEO previously discussed under the decentralization of resources heading. If a single primary (i.e. mission critical) instrument fails in the conventional design with the entire suite of instruments on a single satellite in each plane, a new satellite with the entire suite of new instruments must be immediately launched on a large launch vehicle. Thus, all of the instruments are launched as replacements, even though all but one of the instruments are working properly. This is inherently expensive. If a primary instrument fails in the distributed architecture with the full suite of instruments divided between three satellites in each plane, then only the instruments on the failed satellite need be launched on a smaller spacecraft bus and launch vehicle. Over a long mission, the lower failure compensation (replacement) costs can reduce the total lifecycle *cost* in a distributed satellite system.

#### **18. Multi-Mission Capability**

Distributed Satellite Systems are more amenable to fulfilling multi-mission roles than single satellite deployments. This capability results from the *flexibility* provided by having multiple reconfigurable spacecraft. For example, a DSS design for a spaced-based radar system primarily intended for a ground-moving-target-indication (GMTI) mission may also be tasked to perform secondary missions, such as air-moving-target indication (AMTI) and geolocation. A DSS design maintains the capability to perform these multiple missions due to the inherent ability of the DSS architecture to reconfig-

ure it's array geometry and baseline according to the requirements of each mission.

### **A.3 Allocation of Potential DSS Benefits to Categories**

#### **1. Survivability**

- Decentralization of Resources
- Spatial Distribution
- More Satellites

#### **2. Reduced Cost**

- Smaller, Simpler Satellites
- "Mass Production" of Satellites
- Modular Designs
- Reduced Range
- Reduced Computational Workload
- Redundancy
- Ramped Deployment and Upgrades
- Minimizes Failure Compensation Costs

#### **3. Reduced Time to Initial Operating Capability (IOC)**

- Smaller, Simpler Satellites
- "Mass Production" of Satellites
- Modular Designs

#### **4. Upgradability Ease**

- Modular Designs

#### **5. Improved Revisit Time**

- Spatial Distribution
- Reduced Range
- More Satellites

#### **6. Reduced Power-Aperture**

- Reduced Range
- Separated, Sparse Apertures
- Increased SNR

#### **7. Improved Resolution/Isolation**

- Reduced Range
- Separated, Sparse Apertures
- More Satellites
- Clutter Rejection

#### **8. Improved Rate**

- Multiple Viewing Angles
- Reduced Computational Workload

- More Satellites
  - Task Division
  - Increases SNR
  - Clutter Rejection
- 9. Improved Integrity**
- Spatial Distribution
  - Reduced Range
  - More Satellites
  - Increased SNR
  - Clutter Rejection
- 10. Improved Availability**
- Decentralization of Resources
  - Multiple Viewing Angles
  - More Satellites
  - Redundancy
  - Path Diversity
- 11. Improved Reliability**
- "Mass Production" of Satellites
  - Redundancy
  - Path Diversity
- 12. Lower Failure Compensation Costs**
- Minimization of Failure Compensation Costs
- 13. Improved Visibility/Coverage Geometry**
- Multiple Viewing Angles
  - More Satellites

## **A.4 Generic Metrics Matrix**



TABLE A.1 Generic Metrics Matrix

Metric	Affecting Factor
<b>Isolation</b>	<ul style="list-style-type: none"> <li>• Reduced Range</li> <li>• Separated, Sparse Apertures</li> <li>• More Satellites</li> <li>• Clutter Rejection</li> </ul>
<b>Integrity</b>	<ul style="list-style-type: none"> <li>• Spatial Distribution</li> <li>• Reduced Range</li> <li>• More Satellites</li> <li>• Increased SNR</li> <li>• Clutter Rejection</li> </ul>
<b>Rate</b>	<ul style="list-style-type: none"> <li>• Multiple Viewing Angles</li> <li>• Reduced Computational Workload</li> <li>• More Satellites</li> <li>• Task Division</li> <li>• Increases SNR</li> <li>• Clutter Rejection</li> </ul>
<b>Availability</b>	<ul style="list-style-type: none"> <li>• Decentralization of Resources</li> <li>• Multiple Viewing Angles</li> <li>• More Satellites</li> <li>• Redundancy</li> <li>• Path Diversity</li> </ul>
<b>Performance</b>	<ul style="list-style-type: none"> <li>• Survivability <ul style="list-style-type: none"> <li>•Decentralization of Resources</li> <li>•Spatial Distribution</li> <li>•More Satellites</li> </ul> </li> <li>• Upgradability Ease <ul style="list-style-type: none"> <li>•Modular Designs</li> </ul> </li> <li>• Improved Revisit Times <ul style="list-style-type: none"> <li>•Spatial Distribution</li> </ul> </li> <li>• Reduced Range</li> <li>• More Satellites</li> <li>• Reduced Power-Aperture <ul style="list-style-type: none"> <li>•Reduced Range</li> <li>•Separated, Sparse Apertures</li> <li>•Increased SNR</li> </ul> </li> <li>• Improved Reliability <ul style="list-style-type: none"> <li>•"Mass Production" of Satellites</li> <li>•Redundancy</li> <li>•Path Diversity</li> </ul> </li> <li>• Improved Visibility/Coverage Geometry <ul style="list-style-type: none"> <li>•Multiple Viewing Angles</li> <li>•More Satellites</li> </ul> </li> </ul>

**TABLE A.1** Generic Metrics Matrix

<b>Metric</b>	<b>Affecting Factor</b>
<b>Cost (Cost Per Function)</b>	<ul style="list-style-type: none"><li>• Smaller, Simpler Satellites</li><li>• "Mass Production" of Satellites</li><li>• Modular Designs</li><li>• Reduced Range</li><li>• Reduced Computational Workload</li><li>• Redundancy</li><li>• Ramped Deployment &amp; Upgrades</li><li>• Minimizes Failure Compensation Costs</li></ul>

# Appendix B

## THE LAUNCH VEHICLE SELECTION OPTIMIZATION PROBLEM

This appendix presents an optimization approach to solve the launch vehicle selection problem [Munson & Jilla, 2000]. The selection problem is based on finding the optimal subset of launch vehicles that can deploy all of the satellites in a distributed satellite system at the minimum cost and/or risk; while at the same time adhering to a set of satellite, orbital dynamics, political, and availability constraints. The Launch Vehicle Selection Tool (LST) utilizes this optimization approach to determine the optimal subset of launch vehicles for system deployment. The core of LST is a database that contains information on all of the operational launch vehicles in the market. The information from this database is then combined with the properties of the distributed satellite system to be deployed to create a mathematical formulation of the launch vehicle selection problem as an integer program (IP). The resulting IP is then solved via a branch-and-bound algorithm. This approach was applied to three separate case studies on distributed satellite systems ranging in size from 9 to 288 satellites. In each case, optimal solutions were found for different satellite deployment, replacement, and replenishment scenarios.

### B.1 Introduction

Launch vehicle selection and deployment strategies are an important and expensive issue for satellite system development. Selection of a suboptimal launch scheme can significantly impact cost and schedule of a distributed satellite system program. Satellite devel-

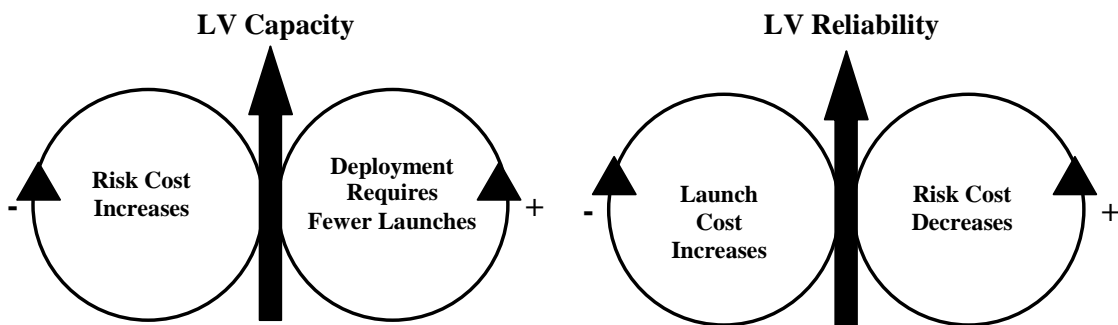
opers currently must choose from over 60 launch vehicle models, with more than twenty new vehicles scheduled to enter service within the next decade. This sheer magnitude of launch options complicates the selection process of a suite of launch vehicles for a distributed satellite system deployment strategy.

Beyond launch costs, it is imperative that distributed satellite system developers pay close attention to vehicle reliabilities. Space launch involves considerable risks. Launch failures such as the 1998 loss of 12 Globalstar communications satellites on board a Zenit 2 vehicle can be financially disastrous. The final insurance claim for this launch mishap totaled approximately \$200 million. The Globalstar failure resulted in millions of dollars in lost revenue for the parent company as well as a one-year schedule slip for market entry. Both of these factors contributed to Globalstar's eventual bankruptcy filing. The potential future introduction of fully reusable launch vehicles designed for high reliability and low per flight costs into the decision matrix will further complicate the vehicle selection process as the number of available options increases.

In the launch vehicle selection problem, an organization would like to deploy a distributed satellite system of  $N$  satellites. The organization must choose from  $n$  different vehicles to launch its satellites. Each launch vehicle has a unique capacity (i.e. number of satellites it can place into orbit), risk, and cost associated with it. The systems engineer's goal is to select the subset of launch vehicles that can deploy all of the satellites in the distributed satellite system at the minimum cost and/or risk; while at the same time adhering to a set of satellite, orbital dynamics, political, and availability constraints.

Figure B.1 illustrates two of the key trade-offs in the launch vehicle selection problem. Selecting launch vehicles with a greater capacity reduces the total number of launches required to deploy the constellation. This, in turn, reduces the total deployment cost of the constellation. However, placing more satellites on a single launch vehicle increases the potential risk to the successful deployment of the full constellation in the case of a failure. Using launch vehicles with a higher reliability can reduce this risk. However, launch vehi-

cles with the best success rates also tend to be the most expensive, driving up the total deployment cost. As will be demonstrated, the Launch Vehicle Selection Tool addresses these key system trades.



**Figure B.1** Trades in the Launch Vehicle (LV) Selection Problem

The primary objective of the Launch Vehicle Selection Tool is to provide distributed satellite system developers with a method for determining the optimal launch deployment strategy. Remember that launch vehicle selection was an important module that impacted both the total lifecycle cost and cost per function metric in the GINA model for each of the three case studies in Chapters 4-6. The Launch Vehicle Selection Tool was the module (i.e. Matlab function) used to solve this problem in each of the three case studies. Given the characteristics of a particular satellite constellation (i.e. the number of satellites, spacecraft dimensions, orbital geometry, etc.), the Launch Vehicle Selection Tool determines the optimal combination of launch vehicles to deploy the system based on three optimization parameters:

1. Minimization of total launch costs,
2. Minimization of program risk, or
3. Minimization of combined program costs and risk.

This appendix outlines the development of the Launch Vehicle Selection Tool and its use to meet this objective. First, the launch vehicle database is introduced. Next, the characteristics of the satellites in the constellation to be deployed were entered into a second database. With this information, preprocessing calculations were executed to determine

how many satellites each launch vehicle can deliver to the desired orbit. Three separate approaches are then presented for modeling the launch vehicle selection problem as an optimization problem. The application of the chosen method, integer programming, to the launch vehicle selection problem is then illustrated. Finally, the launch vehicle selection strategy results from three separate case studies are presented along with the final conclusions.

## **B.2 The LST Database**

Many factors affect the choice of a launch vehicle. While cost and reliability usually receive the most attention, other factors such as physical characteristics and performance attributes eliminate many potential vehicles from consideration. Vehicle restrictions include payload capacity (i.e. mass delivered to a given orbit), fairing dimensions (i.e. height and diameter), integration compatibility with the payload, vehicle performance, political regulations, and launch pad scheduling.

The vehicle database serves as the heart of the Launch Vehicle Selection Tool. The creation of the database is the first step in the process to solve the launch vehicle selection problem. The database was developed to track and store all necessary information for each individual launch vehicle, and consists of 14 parameters for each launch vehicle entry. Design of the database focused on flexibility in order to facilitate the integration of new capabilities into future versions of LST. Future data statistics that may be added to the database include vehicle scheduling and turnaround times, insurance costs, and orbital performance limitations. Table B.1 shows a typical database entry.

The Vehicle Family characteristic identifies the collective name used to describe a line of launch vehicles supported by a particular launch service provider. The Vehicle Class distinguishes the unique vehicle identification. Typically, each launch family contains several distinct vehicles, all based on the same technology and components, yet serving different market demands (i.e. mass, dimensions, orbital inclination, orbital altitude, etc.). The nationality characteristic describes the country of origin for the launch vehicle opera-

**TABLE B.1** Typical Database Entry

<b>Database Parameter</b>	<b>Example</b>
Vehicle Family	Proton
Vehicle Class	Proton K/DM-2
Country	Russia
Provider	ILS/Krunichev
Successful Flights	93
Failures	7
Total Flights	100
Reliability	93%
Mass to Low Earth Orbit	19,760 kg
Vehicle Price	\$82 million
Fairing Volume	191 m <sup>3</sup>
Fairing Height	14.5 m
Fairing Diameter	4.1 m
Weighted Reliability	87.3%

tor organization. As detailed in Section B.4.2, it controls political constraints that might restrict the use of particular vehicles for the deployment of the distributed satellite system. The database currently contains vehicles from eight different countries.

The Flight History parameter chronicles the flight record for each individual launch vehicle and launch vehicle family. The flight record for each vehicle is divided into three parameters detailing the total number of vehicle launches, the number of successful launches, and the number of vehicle failures. The classification of launch as a success or failure is based on the official record from the launch provider. Reliability measures the success rate of a particular launch vehicle. Reliability is a key characteristic calculated by dividing the number of successful vehicle launches by the total number of vehicle flights.

The Mass to Low Earth Orbit characteristic details the maximum mass the vehicle can deliver to a circular orbit 200 kilometers above the Earth's surface. Masses are all in kilograms and rounded off to the nearest ten kilograms. This metric is used to compute the capacity on the number of satellites that can be launched on a single vehicle. The Vehicle Price specifies the amount charged by the service provider to the customer(s) for use of the

entire launch vehicle payload capacity. In some cases, a specific flight might provide the transportation service to more than one satellite client. In this case, the cost per client must be determined. However, the initial model assumes a single client per launch.

The vehicle fairing encases the payload satellite(s) in a shrouded structure that serves as the final stage of a launch vehicle. The volume of the fairing can also set a constraint on the number of satellites that can be deployed into orbit on a single vehicle. Thus, while a particular vehicle might be able to support a certain satellite's mass, it may not be able to accommodate the physical size of the payload. In addition to overall volume constraints, the height and diameter of the fairing also constrain the number and size of the satellites.

The Weighted Reliability characteristic incorporates both the individual and family reliability statistics. This is done to more accurately assess the safety record of the launch vehicle. Thus, a vehicle that has only launched three times without a failure would not receive the same reliability score as a vehicle that has launched 30 times without a failure. The weighted reliability  $R$  is calculated as

$$R = \frac{\frac{Re_V}{(F_V)^{-1}} + \frac{Re_F}{(F_F)^{-1}}}{F_V + F_F} \quad (\text{B.1})$$

where  $Re_V$  and  $Re_F$  represent the reliability of the individual vehicle and the entire family respectively and  $F_V$  and  $F_F$  represent the number of flights made by the individual vehicle and the entire family respectively.

### **B.3 Modeling the Launch Vehicle Selection Problem**

Three separate frameworks were considered for modeling the launch vehicle selection problem – a capacitated network flow model, a network design model, and an integer programming (IP) model. While each approach has its own advantages and disadvantages, the IP formulation was found to best suit the launch vehicle selection problem. This section outlines the decision process that led to this conclusion.



### B.3.1 Capacitated Network Flow Representation

The first modeling approach considered was a capacitated network flow formulation. In capacitated network flow problems, a flow allocation strategy is developed to transport all the units of flow from the source node(s) to the sink node(s) at minimum cost. Figure B.2 illustrates the proposed network flow formulation of the launch vehicle selection problem.

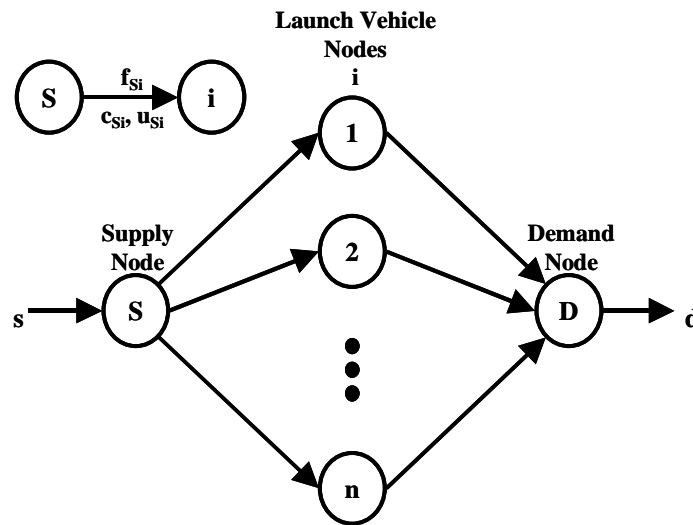


Figure B.2 Capacitated Network Flow Model

This formulation contains three types of nodes – a supply node  $S$ , launch vehicle nodes  $i$ , and a demand node  $D$ . The decision variables  $f_{Si}$  in this formulation represent how many satellites are placed on each launch vehicle  $i$ . One unit of flow corresponds to a single satellite. The amount of flow  $s$  entering the supply node corresponds to the total number of satellites in the constellation to be deployed. The cost  $c_{Si}$  of each arc corresponds to the cost to launch a single satellite on launch vehicle  $i$ . All  $c_{iD}$  arcs have zero cost. The capacity  $u_{Si}$  of each arc represents the maximum number of satellites launch vehicle  $i$  can deliver to the desired orbit. Finally, the demand node serves as a dummy node to ensure that all of the satellites in the distributed satellite system are indeed placed on a launch vehicle. The objective function for this formulation is to select the flow  $f_{Sj}$  that minimizes

the total launch cost of the constellation. Eqn. B.2 contains the mathematical representation for the capacitated network flow model of the launch vehicle selection problem:

$$\begin{aligned}
 & \text{Min } \sum_{i=1}^n c_{Si} f_{Si} \\
 & \text{Subject to } f_{Si} \leq u_{Si} \text{ for all } i \\
 & \quad f_{Si} - f_{iD} = 0 \text{ for all } i \\
 & \quad \sum_{i=1}^n f_{Si} = s \\
 & \quad \sum_{i=1}^n f_{iD} = d \\
 & \quad c_{iD} = 0 \text{ for all } i \\
 & \quad f_{Si}, f_{iD} \geq 0 \text{ for all } i
 \end{aligned} \tag{B.2}$$

The primary advantage of the capacitated network flow formulation is that if all of the supplies, demands, and capacities in the network flow are integral, the solution flow will be integral. This is known as the Integrality Theorem for network flows [Ahuja et al, 1993]. Another advantage of this approach is the availability of specialized algorithms that can quickly solve large minimum cost flow problems with thousands of nodes.

There are two disadvantages with this formulation, however. First, a different node  $i$  must be placed in the network not only for each model of each launch vehicle, but also for every available serial vehicle. For example, if there are five Delta II 7925 rockets available in one year, then each of these vehicles requires its own node in the network formulation. Thus, the size of the network becomes excessively large when every serial production of every launch vehicle model in every launch vehicle family is considered. The sheer size of the resulting model makes it time-consuming to formulate and difficult to revise.

The second drawback to this formulation is the fact that the cost of launching a given number of satellites on a rocket does not scale linearly with the number of satellites placed

on the rocket. For example, assume that a vehicle costs \$50 million and can carry five satellites to orbit. The total cost to use that rocket will be \$50 million whether one, two, three, four, or five satellites are placed on it. Thus, if  $c_{S_i}$  is calculated by dividing the cost of launch vehicle  $i$  by the maximum number of satellites it can deploy, then the representation of launch vehicle cost as  $c_{S_i}f_{S_i}$  is incorrect if the flow through an arc into launch vehicle node  $i$  is less than the capacity of that arc. For these reasons, a capacitated network flow formulation was not chosen for the launch vehicle selection problem.

### B.3.2 Network Design Formulation

The second modeling approach considered was a network design formulation. In network design problems, specific arcs are chosen to create a network topology that meets all flow requirements at minimum cost. Figure B.3 illustrates the proposed network design formulation of the launch vehicle selection problem.

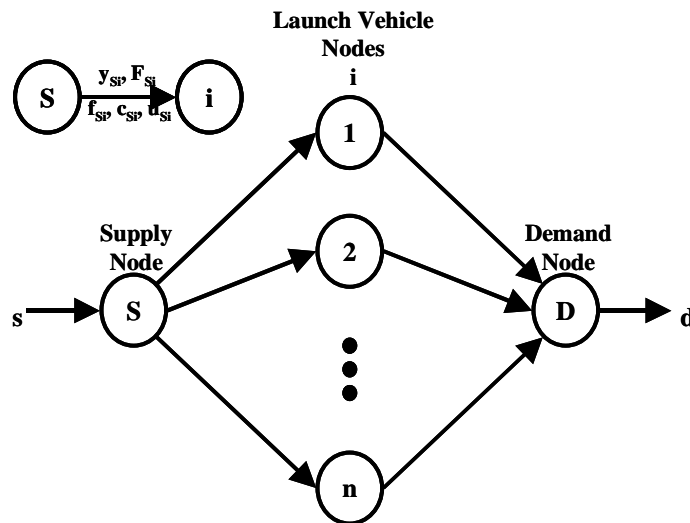


Figure B.3 Network Design Model

This formulation again contains three types of nodes – a supply node  $S$ , launch vehicle nodes  $i$ , and a demand node  $D$ . The decision variables in this model include a binary variable (i.e. 0 or 1)  $y_{S_i}$  that denotes whether or not arc  $S_i$  is placed in the network (i.e. whether

or not to use launch vehicle  $i$ ) at a fixed cost  $F_{Si}$ , which is independent of the amount of flow sent through the arc. If a given arc is included in the network topology, then the second decision variable  $f_{Si}$  determines the amount of flow (i.e. number of satellites) placed through that arc. If all arc flow costs (i.e.  $c_{Si}$  and  $c_{iD}$ ) are set to zero, then the objective function for this formulation simplifies to selecting a set of arcs  $y_{Si}$  to create a network that minimizes the total launch cost of the constellation while satisfying all flow constraints. Eqn. B.3 contains the mathematical representation for the network design model of the launch vehicle selection problem:

$$\begin{aligned}
 & \text{Min } \sum_{i=1}^n (c_{Si}f_{Si} + y_{Si}F_{Si}) \\
 & \text{Subject to } f_{Si} \leq u_{Si} \text{ for all } i \\
 & \quad f_{Si} - f_{iD} = 0 \text{ for all } i \\
 & \quad \sum_{i=1}^n f_{Si} = s \\
 & \quad \sum_{i=1}^n f_{iD} = d \\
 & \quad c_{Si} = 0 \text{ for all } i \\
 & \quad f_{Si}, f_{iD} \geq 0 \text{ for all } i \\
 & \quad 0 \leq y_{Si} \leq 1 \text{ and } y_{Si} \text{ integer for all } i
 \end{aligned} \tag{B.3}$$

The advantage of this formulation is that it successfully addresses the fact that launch costs do not scale linearly with the number of satellites placed on a launch vehicle. The disadvantage to this formulation, however, is that the algorithms for solving network design problems are much slower than corresponding solution algorithms for network flow problems, requiring minutes to hours to solve even the simplest cases. For this reason, the network design formulation was also not chosen to model the launch vehicle selection problem.

### B.3.3 Integer Programming

The third modeling technique considered was integer programming. An integer program is a linear program in which some or all of the decision variables are required to be integers. Integer programming (IP) offers the advantage of being a powerful modeling tool for problems with complex constraints. However, certain instances of integer programs can be very hard to solve to optimality. In light of the ease and power of this modeling technique, integer programming was selected to formulate the launch vehicle selection problem. The next section outlines the integer programming formulation in detail.

## B.4 IP Formulation of the Launch Vehicle Selection Problem

### B.4.1 Decision Variables and Objective Function

The decision variables  $x_i$  in the integer programming formulation of the launch vehicle selection problem represent how many of each launch vehicle  $i$  should be used to deploy the distributed satellite system. The objective function is to select a suite of launch vehicles to deploy the distributed satellite system at minimum cost:

$$\text{Min } \sum_{i=1}^n C_i x_i \quad (\text{B.4})$$

where  $C_i$  is the cost of using launch vehicle  $i$ .

Table B.2 illustrates the three different methods for calculating the cost  $C_i$  of launch vehicle  $i$ . The selection of the type of cost incorporated into the objective function will depend on the priorities of the organization developing the distributed satellite system. If the program budget is tight and the priority is to minimize the total absolute launch cost, then  $C_i$  is the actual cost  $C_{LVi}$  that the launch vehicle provider charges the customer for rocket  $i$ .

In many real world programs, however, perceived risk can be just as important as actual costs. For example, a commercial entity that wants to deploy a high cost, high revenue

**TABLE B.2** Different Objective Function Strategies

Organization Priority	Corresponding Objective Function
Minimize the Absolute Launch Cost	$\text{Min} \sum C_{LVi} x_i$
Minimize the Total Program Risk	$\text{Min} \sum C_{Ri} x_i$
Balance Total Cost and Total Risk	$\text{Min} \sum x_i (C_{LVi} + C_{Ri})$

telecommunications satellite might be willing to spend several million dollars more to secure a launch vehicle with a 99% success rate than to use a less expensive launch vehicle with a success rate of only 75%. If the objective is to minimize the total deployment risk of the constellation, then  $C_i$  is the risk cost of using launch vehicle  $i$ . Risk cost  $C_{Ri}$  is calculated as the product of the probability of failure  $P_{LVFi}$  of launch vehicle  $i$  with the cost of failure  $C_{LVFi}$ , which is the sum of the cost of the launch vehicle and all of the satellites it is carrying:

$$C_{Ri} = P_{LVFi} C_{LVFi} \quad (\text{B.5})$$

Finally, if both absolute cost and risk are important, then  $C_i$  is the sum of  $C_{LVFi}$  and  $C_{Ri}$ . The objective function in this case captures the trade-off between total absolute cost and total potential risk. These three objective functions will likely produce different "optimal" launch vehicle selection strategies, and the case studies at the end of this appendix investigate this phenomenon.

### B.4.2 Constraints

Many different constraints must be included in the formulation of the launch vehicle selection problem. This is where the strength of the integer programming approach becomes apparent.

#### Total Satellite Deployment Constraint

As computed in the database, each launch vehicle can deploy a different given number of satellites, depending upon the mass and volume of the satellites in their stowed configura-

tion as well the constellation orbital altitude and inclination. This is captured by the launch vehicle satellite capacity  $K_i$ . The total number of satellites deployed in the IP solution must be greater than or equal to the total number of satellites  $N$  in the constellation:

$$\sum_{i=1}^n K_i x_i \geq N \quad (\text{B.6})$$

The reason a greater than or equal to sign is used instead of an equal sign is because it might be more cost-effective to launch more satellites on a cheaper rocket than fewer satellites on a more expensive rocket.

### Orbital Dynamics Constraint

The orbital dynamics of the distributed satellite system being deployed impose two separate constraints on the integer program. In a typical constellation, the satellites are evenly divided between a given number of orbital planes. The orbital inclination and the longitude of the ascending node in the rotating coordinate frame define unique orbital planes. Once a satellite is deployed within a given orbital plane, it becomes very expensive in terms of fuel consumption to move the satellite into a different orbital plane. Doing so decreases the useful life of the satellite. This fact may be incorporated into the integer program by imposing the constraint that the minimum number of launches to deploy the distributed satellite system cannot be less than the total number of orbital planes  $n_{OP}$  in the constellation:

$$\sum_{i=1}^n x_i \geq n_{OP} \quad (\text{B.7})$$

This ensures that satellites will be deployed in each of the orbital planes. Additionally, the capacity  $K_i$  of each launch vehicle is taken to be the minimum of the mass/volume limited capacity and the total number of satellites in each orbital plane to ensure that more satellites are not delivered to a single orbital plane than required.

### Political Constraints

Numerous political constraints arise when deploying a satellite constellation, and this is where the strength of IP modeling truly demonstrates itself. Political constraints are a function of the type of organization operating the satellite constellation and the laws of the country in which that organization is based. For example, federal law mandates that all U.S. government satellites can only be launched by U.S. rockets. The laws for private commercial companies become more complicated. Currently, the U.S. government has imposed quotas limiting the use of Russian and Ukrainian launch vehicles by American satellite companies. The intent is to protect American aerospace companies from being driven out of the market by the cheaper, government-subsidized launch vehicles from these countries. The U.S. government also often places sanctions on countries' industries, including their launch vehicle industry, as a foreign policy tool. For example, several export licenses for the launch of U.S. manufactured satellites aboard Chinese launch vehicles have been halted due to recent political events beyond the control of the buyers and sellers in the aerospace industry. All of these different ceilings on the maximum number of launch vehicles that may be used from different countries may be modeled as:

$$\sum_{i=1}^m x_{ij} \leq Q_j \text{ for all } j \quad (\text{B.8})$$

where  $m$  is the total number of launch vehicle models from country  $j$  and  $Q_j$  represents the total number of launch vehicles from country  $j$  that are allowed by law to be used in a given year.

While federal regulations often impose upper bounds on the total number of launch vehicles that may be used from a particular country, other situations often place lower bounds on the number of launch vehicles that must be used from a single country. Consider an international consortium that is developing a global telecommunication satellite network. If, for example, the consortium is made up of American, European, and Japanese companies, it is not uncommon that the consortium will agree to use a given number of launch



vehicles from companies located in these three countries before considering vehicles from any other country. Such launch vehicle floor constraints may be modeled in the following manner:

$$\sum_{i=1}^m x_{ij} \geq M_j \text{ for all } j \quad (\text{B.9})$$

where  $M_j$  represents the minimum number of launch vehicles that must be selected from country  $j$ .

### Availability Constraints

An upper bound  $A_i$  is placed on the maximum number of each particular launch vehicle model  $i$  that may be used:

$$x_i \leq A_i \text{ for all } i \quad (\text{B.10})$$

This upper bound arises from the fact that only a limited number of each launch vehicle is manufactured each year. Availability is further limited by the fact that many launch vehicles are already spoken for by contracts signed many years earlier.

### Integrality Constraints

All decision variables  $x_i$  in the solution to the IP are constrained to be integers:

$$x_i \text{ integer for all } i \quad (\text{B.11})$$

In reality, only integral numbers of launch vehicles may be used – there is no such thing as launching 1.4 rockets.

### Non-Negativity Constraints

As in most minimization problems, negative values for the decision variables are not allowed:

$$x_i \geq 0 \text{ for all } i \quad (\text{B.12})$$

### B.4.3 IP Model Formulation Summary

Table B.3 shows the complete formulation of the launch vehicle selection problem as an integer program. This formulation contains 43 decision variables and – including upper and lower bounds, not including integrality constraints, and depending upon the number of political constraints – approximately 100 constraints.

**TABLE B.3** IP Formulation of the Launch Vehicle Selection Problem

Mathematical Formulation	Explanation
Objective Function	
Min $\sum_{i=1}^n C_i x_i$	Minimize Costs
Subject to	
$\sum_{i=1}^n K_i x_i \geq N$	Total Satellite Deployment Constraint
$\sum_{i=1}^n x_i \geq n_{OP}$	Orbital Dynamics Constraint
$\sum_{i=1}^m x_{ij} \leq Q_j$ for all $j$	Federal Law/Quotas/Sanctions Political Constraints
$\sum_{i=1}^m x_{ij} \geq M_j$ for all $j$	International Consortium Political Constraints
$x_i \leq A_i$ for all $i$	Availability Constraints
$x_i$ integer for all $i$	Integrality Constraints
$x_i \geq 0$ for all $i$	Non-Negativity Constraints

## B.5 Integer Program Solution Techniques

Once the launch vehicle selection problem has been modeled as an integer program, an optimization algorithm must be applied to solve the problem. Integer programs generally fall into two subclasses, mixed integer models and pure integer models. A mixed integer problem is a mathematical program requiring at least one, but not all, of the decision variables to be integers. A pure integer problem involves restricting all decision variables to take on integer values. The launch vehicle selection problem is a pure integer problem.

While the formulation of an integer program might closely resemble a typical linear program (LP) with continuous variables, the mathematical solution of an IP model is generally more rigorous than a similar LP model. Compared to LP models with thousands of variables and constraints, it is not uncommon to build an IP model of similar magnitude that cannot be solved in a reasonable amount of time.

Unlike the very successful LP simplex algorithm, the search for faster IP algorithms still continues. Several solution methodologies have been developed with varying degrees of success and are highly dependent on the particular application. The most common and successful method used to solve integer programs has been the branch-and-bound technique. The first step in the branch-and-bound algorithm involves relaxing the integrality constraints and solving the integer program as a linear program. Calculation of upper and lower bounds on the objective function narrow the feasible region and reduce the amount of enumeration required. Through a process called branching, the initial problem is divided into two new subproblems. New bounds are calculated for the subproblems and the process continues toward the optimal solution. This process forms an enumeration tree. The enumeration tree shows a progression through feasible solutions and, through a process called pruning, infeasible or non-optimal solutions are eliminated or cut from the tree. The launch vehicle selection problem IP was solved via the branch-and-bound algorithm.

## **B.6 Case Studies**

The case studies represent the application of the selected modeling and solution techniques to real world launch vehicle selection problems. Three separate case studies of three very different distributed satellite systems were selected to demonstrate the versatility of the integer programming approach. Table B.4 lists the characteristics of each of the three case study distributed satellite systems. It should be stressed that the constellations in these three case studies are similar, but not identical to, the constellations on which they are based.

**TABLE B.4** Characteristics of the Three Case Studies

	<b>Weather Constellation</b>	<b>Telephony Constellation</b>	<b>Broadband Constellation</b>
Case Study #	1	2	3
Based On	NPOESS	Iridium	Teledesic
Ownership	U.S. Government	U.S. Company	Intl. Consortium
Total # Satellites	9	72	288
# Orbital Planes	3	6	12
# Satellites Per Plane	3	12	24
Satellite Mass	971 kg	690 kg	500 kg
Satellite Cost	\$75 M	\$25 M	\$15 M

The first case study is for a nine-satellite weather surveillance constellation based on a proposed configuration for the National Polar-Orbiting Operational Environmental Satellite System (NPOESS). NPOESS is currently in the design phase and intends to replace with a single system the current civilian and military constellations of low orbiting weather satellites now in use. This constellation will be owned by the U.S. government, and therefore only U.S. launch vehicles may be used in its deployment. Each satellite carries several scientific instruments, making the satellites fairly large and expensive.

The second case study, based on the recently deployed Iridium system, is a 72 satellite (i.e. 66 operational, six in-orbit spares) constellation to be used primarily for global paging and telephone communications. As modeled here, the system is owned by a U.S. company, and must therefore obey all Federal laws, including the quotas placed on launch vehicles from various countries. The satellites in this constellation are smaller and cheaper than the satellites in the first case study. This has the implication of increasing the launch vehicle capacities while lowering the associated risk costs.

The third case study is for a 288 satellite constellation to be used for global broadband communications. This case study is based on an old version of the proposed Teledesic system. Because this system is modeled as being owned by an international consortium,

launch vehicles from each of the member countries in the consortium must be used to deploy the satellites. When so many identical satellites are manufactured for a single constellation, learning curve savings and satellite redundancy lead to the use of cheaper, smaller satellites.

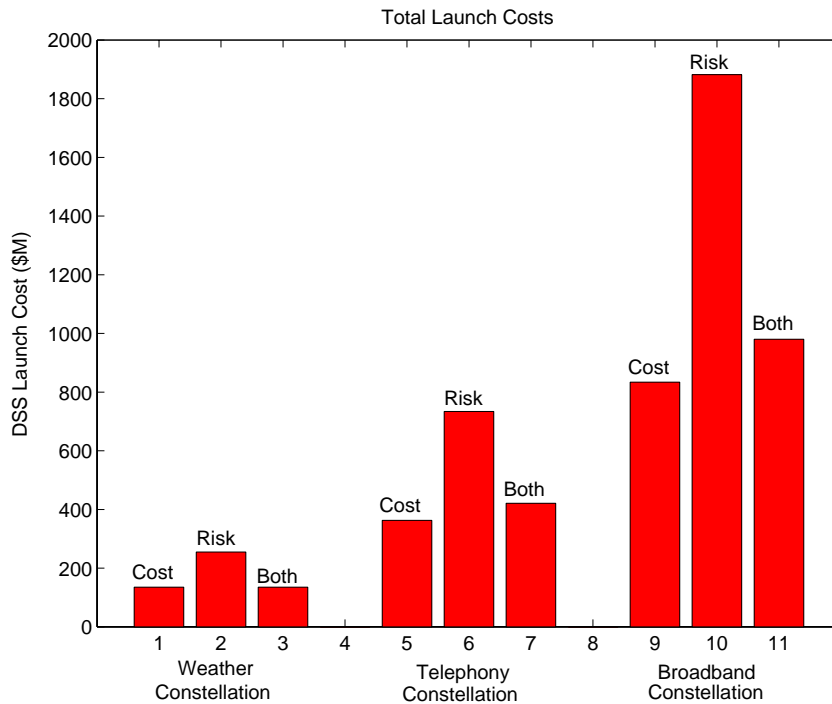
## **B.7 Results**

### **B.7.1 Test Matrix**

A test matrix was created for each case study. All three objective functions listed in Table B.2 were applied to each of the three constellations. For each objective function, six separate cases were run:

1. LP Relaxation for full constellation deployment.
2. IP Solution for full constellation deployment.
3. LP Relaxation for single satellite replacement.
4. IP Solution for single satellite replacement.
5. LP Relaxation for multiple satellite replenishment.
6. IP Solution for multiple satellite replenishment.

The solution to the LP relaxation – in which the integer constraints in Eqn. B.11 and Table B.3 are removed – for full constellation deployment places a lower bound on the minimum total cost for which all of the satellites may be launched. This is because the solution to the LP relaxation can never be worse than the solution to the more highly constrained equivalent IP problem. Next, the IP solution is found for the same problem. In addition to the full constellation deployment runs, optimizations were also executed to determine the best replacement strategies for launching a single satellite in the case of an in-orbit failure as well as a multi-satellite replenishment strategy for Case Studies Two and Three. The replenishment strategy lofts three satellites at a time in Case Study Two and six satellites at a time in Case Study Three. Table B.5 through Table B.7 list the results for each of these optimization runs. Figure B.4 plots the full constellation deployment total



**Figure B.4** Total Launch Costs for Different Objective Functions in the Three Case Studies

launch cost of the optimized launch vehicle deployment strategy for each objective function in each case study.

### B.7.2 Case Study 1 – Weather Constellation

The results for case study one in Table B.5 are straightforward and consistent. The IP solution matches the LP relaxation solution for all three objective functions and all solver tolerance settings. Absolute launch cost is minimized by using three Delta II 7420 launch vehicles to deploy the nine weather satellites. For the risk minimization objective function, the algorithm chose to use three Atlas 2 launch vehicles, which hold a 100% reliable launch record at the time of writing. Finally, cost and risk are best balanced through the selection of three Delta II 7420 rockets. For these three objective functions, the optimal launch vehicles for single satellite replacements are the Taurus, Atlas 2, and Taurus again.

**TABLE B.5** Case Study 1 Results – 9 Satellite Weather Constellation

	<b>Minimize Absolute Launch Cost</b>	<b>Minimize Total Program Risk</b>	<b>Minimize Total Cost + Total Risk</b>
<b>LP Relaxation – Full Deployment</b>			
Objective Function Value	\$135 M	\$0 M	\$157.1 M
# Launch Vehicles Used	3	3	3
<b>IP – Full Deployment</b>			
Objective Function Value	\$135 M	\$0 M	\$157.1 M
% from Lower Bond	0%	0%	0%
# Launch Vehicles Used	3	3	3
<b>LP Relaxation – Single Satellite</b>			
Objective Function Value	\$85 M	\$0 M	\$18 M
# Launch Vehicles Used	1	1	1
<b>IP – Single Satellite</b>			
Objective Function Value	\$85 M	\$0 M	\$18 M
% from Lower Bond	0%	0%	0%
# Launch Vehicles Used	1	1	1

### B.7.3 Case Study 2 – Telephony Constellation

Table B.6 illustrates the results for case study two. The cost minimization objective function value of the IP solution differs from the LP relaxation solution by 3.7%, with eight separate launches needed to deploy the constellation. The risk minimization strategy produces drastically different results. The IP solutions to this objective use 18 launches to deploy the constellation, with the objective value differing from the LP relaxation bound by 2.7%. The best launch vehicle selection strategy to balance cost and risk uses nine launch vehicles, and comes within 3.2% of the LP relaxation bound. The optimal launch vehicles for single satellite replacements and three satellite replenishments for the three objective functions are the Dnepr, Atlas 2, and Dnepr respectively. Figure B.4 illustrates the drastic increase in total launch cost when the selection priority shifts from cost minimization to risk minimization.

**TABLE B.6** Case Study 2 Results – 72 Satellite Telephony Constellation

	<b>Minimize Absolute Launch Cost</b>	<b>Minimize Total Program Risk</b>	<b>Minimize Total Cost + Total Risk</b>
<b>LP Relaxation – Full Deployment</b>			
Objective Function Value	\$350 M	\$14.6 M	\$464.7 M
# Launch Vehicles Used	7.5	16.4	8.7
<b>IP – Full Deployment</b>			
Objective Function Value	\$363 M	\$15 M	\$479.8 M
% from Lower Bond	3.7%	2.7%	3.2%
# Launch Vehicles Used	8	18	9
<b>LP Relaxation – Single Satellite</b>			
Objective Function Value	\$8 M	\$0 M	\$9.1 M
# Launch Vehicles Used	1	1	1
<b>IP – Single Satellite</b>			
Objective Function Value	\$8 M	\$0 M	\$9.1 M
% from Lower Bond	0%	0%	0%
# Launch Vehicles Used	1	1	1
<b>LP Relaxation – Multiple Satellites</b>			
Objective Function Value	\$8 M	\$0 M	\$10.7 M
# Launch Vehicles Used	1	1	1
<b>IP – Multiple Satellites</b>			
Objective Function Value	\$8 M	\$0 M	\$10.7 M
% from Lower Bond	0%	0%	0%
# Launch Vehicles Used	1	1	1

### B.7.4 Case Study 3 – Broadband Constellation

Table B.7 illustrates the results for case study three. For the cost minimization objective, the constellation deployment IP solution reaches the LP relaxation bound to within 0.4% using 24 launches. The IP solution to the risk minimization objective uses 41 launch vehicles to achieve an objective function value within 0.9% of the LP relaxation bound. Finally, the LP and IP solutions to the balanced cost and risk objective function are identical, with 27 launches selected to deploy all 288 satellites. Again, Figure B.4 illustrates a dramatic increase in total launch cost when the selection priority shifts from cost minimization to risk minimization. Using an objective function that balances both cost and risk



leads to a vehicle selection strategy only slightly more expensive than the minimum cost strategy. The optimal launch vehicles for single satellite replacements and six satellite replenishments for the three objective functions are again the Dnepr, Atlas 2, and Dnepr.

**TABLE B.7** Case Study 3 Results – 288 Satellite Broadband Constellation

	<b>Minimize Absolute Launch Cost</b>	<b>Minimize Total Program Risk</b>	<b>Minimize Total Cost + Total Risk</b>
<b>LP Relaxation – Full Deployment</b>			
Objective Function Value	\$830.8 M	\$158.1 M	\$1390 M
# Launch Vehicles Used	22.6	42.2	27
<b>IP – Full Deployment</b>			
Objective Function Value	\$834 M	\$159.5 M	\$1390 M
% from Lower Bond	0.4%	0.9%	0%
# Launch Vehicles Used	24	41	27
<b>LP Relaxation – Single Satellite</b>			
Objective Function Value	\$8 M	\$0 M	\$8.7 M
# Launch Vehicles Used	1	1	1
<b>IP – Single Satellite</b>			
Objective Function Value	\$8 M	\$0 M	\$8.7 M
% from Lower Bond	0%	0%	0%
# Launch Vehicles Used	1	1	1
<b>LP Relaxation – Multiple Satellites</b>			
Objective Function Value	\$8 M	\$0 M	\$11.1 M
# Launch Vehicles Used	1	1	1
<b>IP – Multiple Satellites</b>			
Objective Function Value	\$8 M	\$0 M	\$11.1 M
% from Lower Bond	0%	0%	0%
# Launch Vehicles Used	1	1	1

## B.8 Conclusions

To summarize, this appendix has outlined the motivation behind the launch vehicle selection problem, detailed the creation of the launch vehicle database needed to solve this problem, and summarized the results from three case studies. It also explained the IP

modeling approach used to formulate the problem in a mathematical framework and introduced the branch-and-bound optimization algorithm as the solution technique.

Use of an optimization solver linked to a launch vehicle database provides tremendous versatility to the solution of the launch vehicle selection problem. This is especially important when working in the dynamic space launch market. New launch vehicles with differing capabilities and costs can be introduced, failures can ground operational launch vehicles, or quotas and export license blocks can be placed on vehicles from different countries by the whims of Congress and the State Department. The Launch Vehicle Selection Tool presented in this appendix can be used to quickly and effectively find a new, optimized launch vehicle selection strategy for the full deployment, replacement, or replenishment of the satellites within a distributed satellite system.

This versatility was demonstrated by the three case studies presented in this appendix. Each case study covered a constellation with very different satellite deployment requirements and constraints. Three separate objective functions were applied to each case study to capture different deployment strategies with respect to cost and risk. In each case, regardless of the objective function, number of satellites, political constraints, etc.; optimal solutions were found that come close to the LP relaxation lower bound. Thus, the use of IP modeling with the branch-and-bound optimization algorithm was found to be a robust technique for solving the launch vehicle selection problem.