

**A MULTI-OBJECTIVE STOCHASTIC APPROACH TO
COMBINATORIAL TECHNOLOGY SPACE
EXPLORATION**

A Thesis
Presented to
The Academic Faculty

by

Chirag B. Patel

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
August 2009

A MULTI-OBJECTIVE STOCHASTIC APPROACH TO COMBINATORIAL TECHNOLOGY SPACE EXPLORATION

Approved by:

Dr. Dimitri N. Mavris, Adviser
Professor
Georgia Institute of Technology

Dr. Daniel P. Schrage
Professor
Georgia Institute of Technology

Dr. Michelle R. Kirby
Research Engineer II
Georgia Institute of Technology

Dr. Brian J. German
Assistant Professor
Georgia Institute of Technology

Dr. Frederic Villeneuve
Senior Engineer
Siemens Power Generation, Inc.

Ms. Antje Lembcke
Manager, Gas Turbine Technology
Development
Siemens Power Generation, Inc.

Date Approved: 15 May 2009

ACKNOWLEDGEMENTS

I'd like to extend my sincere gratitude to many individuals who have helped me achieve this success. My advisor, Dr. Dimitri Mavris has provided the encouragement and support for this endeavor. He has provided me with various opportunities and continually motivated me through my years at Georgia Tech. Doc, I feel fortunate to have you as my mentor and friend. You've truly cared about my success and well being. I'd like to thank Dr. Daniel Schrage for his valuable comments and encouragement for my research. Dr. Michelle Kirby has brought her considerable experience in technology prioritization to my thesis committee and I feel privileged to have her by my side. This research would not have been the same without her inputs and feedback, especially with the aircraft technology problem. Dr. Brian German has been a good guide and a friend who has always provided sound advice. His feedback was invaluable in refining the probabilistic Pareto optimization technique. I'm thankful to Ms. Antje Lembcke and Dr. Frederic Villeneuve for bringing in the industry perspective to my thesis. Ms. Lambcke's inputs have been invaluable in restructuring and refining the technology exploration process. Dr. Villeneuve has provided valuable help with proof reading the thesis and a good sounding board for ideas. They, along with Doc, have been instrumental in my recent employment with Siemens and I'm truly grateful for that.

I've been fortunate to be a part of ASDL that has provided me with a wealth of experience, contact and life long friends. Among many friends, I'd like to give special thanks to Eunsuk Yang and Dr. Sriram Rallabhandi for their help and friendship over the years at Tech. Suraj, Rama, Suresh, Manuj, Henry, Samson, Samer, Yuan and others have made my time in Atlanta very enjoyable and I will cherish that forever.

I would like to thank my parents, Nayana and Babubhai Patel for their love and unwavering support throughout these years. They are the one to encourage me to take this path and I'm eternally grateful for that.

Last but not least, I would like to thank my wife Tanha and daughter Janvi for their love and patience over the years. Tanha, your love and help, especially at the home front, kept me going through this journey and Janvi's cheerful exuberance helped me reach the destination. Without you, I would not have accomplished this. I love you both.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iii
LIST OF TABLES	xi
LIST OF FIGURES	xii
SUMMARY	xv
I INTRODUCTION	1
1.1 Motivation	2
1.1.1 Requirements and Resource Matching	2
1.1.2 Knowledge Based Development and Acquisition	5
1.1.3 Immature Technologies and their Impact	8
1.1.4 Selecting the Right Technologies	10
1.2 The Technology Selection Problem	12
1.2.1 Core Problem	12
1.2.2 Multi-Objective Design Space	13
1.2.3 Technological Uncertainties	13
1.3 Research Objectives	14
1.3.1 Research Questions	15
II STATE OF ART IN TECHNOLOGY SELECTION	16
2.1 Technology Identification Evaluation and Selection (TIES)	16
2.1.1 Advantages and Shortcomings	18
2.2 Technology Metric Assessment and Tracking (TMAT)	19
2.2.1 Advantages and Shortcomings	20
2.3 Strategic Prioritization Process (SP2)	20
2.3.1 Advantages and Shortcomings	21
2.4 Strategic Assessment of Risk and Technologies (START)	22
2.4.1 Decision Tree Assessment	23
2.4.2 Inference Nets	24

2.4.3	Post-optimality Analysis	25
2.5	Other Techniques for Technology Assessment and Portfolio Planning	26
2.5.1	Strategic Technology Assessment	26
2.5.2	Portfolio Planning	27
2.6	Observations	28
2.6.1	Useful Techniques	30
2.7	Summary	31
III	ALGORITHMS FOR TECHNOLOGY SELECTION	33
3.1	Technology Selection and the <i>Knapsack</i> Problem	33
3.1.1	Benchmark Knapsack Problem	35
3.2	Approximate Algorithms	36
3.2.1	Greedy Algorithms	37
3.2.2	Monte-Carlo Methods	42
3.3	Exact Algorithms	44
3.3.1	Branch and Bound Algorithms	45
3.3.2	Dynamic Programming Algorithms	49
3.4	Investigative Techniques	53
3.4.1	One-On One-Off	53
3.4.2	Design of Experiments	56
3.5	Advisor for Technology Selection Techniques	59
3.6	Summary	63
IV	MULTI-OBJECTIVE DECISION MAKING	64
4.1	MODM Approaches	64
4.1.1	<i>A Priori</i> Preference Articulation	66
4.1.2	<i>Progressive</i> Preference Articulation	72
4.1.3	<i>A Posteriori</i> Preference Articulation	72
4.2	Pareto Optimality	74
4.3	Challenges with <i>A Posteriori</i> Preference Articulation	76

4.3.1	Reducing the Dimensionality of Pareto Frontier	77
4.3.2	Search for Pareto Optimal Solutions	88
4.4	Summary	90
V	EVOLUTIONARY ALGORITHMS FOR PARETO OPTIMIZATION	92
5.1	Evolutionary Computation	92
5.1.1	Why Evolutionary Algorithms?	93
5.1.2	No Free Lunch Theorems	95
5.2	Pareto Optimization Using EAs	96
5.2.1	Fitness Assignment	97
5.2.2	Distribution Along the Surface	98
5.2.3	Elitism	99
5.3	Fonseca and Fleming GA (FFGA)	100
5.3.1	Advantages and Shortcomings	102
5.4	Non-dominated Sorting GA I & II (NSGA I & II)	102
5.4.1	Advantages and Shortcomings	104
5.5	Niched Pareto Genetic Algorithm (NPGA)	104
5.5.1	Pareto-Domination Tournaments	105
5.5.2	Sharing	106
5.5.3	Advantages and Shortcomings	107
5.6	Strength Pareto Evolutionary Algorithm I & II (SPEA I & II)	107
5.6.1	Advantages and Shortcomings	109
5.7	Comparing NPGA and SPEA II	110
5.7.1	Criteria for Performance Comparison	111
5.7.2	Implementation of Algorithms	112
5.7.3	Simulation Results	114
5.8	Efficacy of SPEA II	117
5.8.1	Effect of Changing Algorithmic Parameters	117
5.8.2	Comparing SPEA II with Random Search	120

5.9	Summary	122
VI	PROBABILISTIC TECHNOLOGY SELECTION	123
6.1	Probabilistic Design	123
6.2	Technological Uncertainties	126
6.2.1	Epistemic Uncertainty	126
6.2.2	Uncertainty Representation	128
6.3	Probabilistic Analysis	131
6.3.1	Convolution	132
6.3.2	Mean Value Methods	133
6.3.3	Monte Carlo Simulation (MCS)	135
6.3.4	Probabilistic Analysis for Complex Functions	137
6.4	Probabilistic Optimization	140
6.5	Proposed Probabilistic Technology Selection Approach	141
6.5.1	Joint and Marginal Probability Distributions	142
6.5.2	Probabilistic Pareto Layers	146
6.6	Probabilistic Pareto Optimization	148
6.6.1	Fitness Calculation	148
6.6.2	Archiving	148
6.7	Validating the Approach on a Knapsack Problem	149
6.7.1	Probabilistic Knapsack Problem	149
6.7.2	Deterministic Results	150
6.7.3	Probabilistic Results	151
6.7.4	Result Comparison	153
6.8	Summary	158
VII	TECHNOLOGY CONSTRAINTS	160
7.1	Types of Technology Interactions	160
7.1.1	Simple Interactions	161
7.1.2	Non-Simple Interactions	163

7.2	Graph Theory Connection	164
7.2.1	Counting Permissible Technology Combinations	165
7.2.2	Average Number of Independent Sets	168
7.3	Enumeration with Backtracking	169
7.4	Enabling Technologies	171
7.5	Technology Constraints with Evolutionary Algorithms	172
7.5.1	Soft Constraints	173
7.5.2	Hard Constraints	174
7.6	Summary	175
VIII	PARETO OPTIMIZATION AND SELECTION OF TECHNOLOGIES .	176
8.1	Proposed Method	176
8.2	Problem Formulation	179
8.3	Collecting Technology Data	182
8.4	Estimating Computational Complexity	183
8.5	Search for True Pareto Front or Layers	184
8.5.1	Enumerate Permissible Combinations	184
8.5.2	Evaluate Deterministically or Probabilistically	185
8.5.3	Extract True Pareto Front or Layers	186
8.5.4	Reduce Dimensionality?	187
8.5.5	Reduce Dimensions with k -EMOSS Approach	188
8.5.6	Extract Pareto Front or Layers for Selected Objectives	189
8.6	Pareto Optimization	190
8.6.1	Reduce Dimensions?	190
8.6.2	Deterministic Pareto-Optimization	191
8.6.3	Reduce Dimensions with k -EMOSS Approach	192
8.6.4	Deterministic or Probabilistic Pareto-Optimization	192
8.7	Exploring and Selecting the Technologies	193
8.8	Summary	194

IX	EXPLORING TECHNOLOGIES FOR A COMMERCIAL AIRCRAFT	196
9.1	Aircraft Technology Problem Formulation	196
9.1.1	Baseline	197
9.1.2	Modeling and Simulation	198
9.2	Technology Data	200
9.3	Complexity of Technology Graph	202
9.4	Reduce Dimensionality?	203
9.5	Deterministic Pareto Optimization	203
9.6	Dimensionality Reduction	204
9.7	Probabilistic Pareto Optimization	207
9.8	Exploring and Selecting the Technologies	210
9.8.1	Scatter Plots	213
9.8.2	Clustering	217
9.8.3	Strategies for Visual Exploration and Decision Making . . .	220
9.9	Summary	226
X	CONCLUSIONS	228
10.1	Summary of Contributions	228
10.1.1	Technology Selection Advisor	228
10.1.2	Multi-Objective Technology Decisions	229
10.1.3	Probabilistic Pareto Optimization	230
10.1.4	Technology Incompatibilities	231
10.1.5	Combinatorial Technology Space Exploration	232
10.2	Recommendations and Future Work	233
10.2.1	Ideas for Further Research	234
10.3	In Closing	235
	REFERENCES	236
	VITA	249

LIST OF TABLES

1	Technology Readiness Levels	8
2	Time Complexity of Technology Combinatorial Space	13
3	Method Comparison	29
4	Legend for Method Comparison	29
5	Example Knapsack Problem	36
6	Example Multi-Objective Problem	78
7	Correlation Matrix \mathbf{R}	79
8	Eigenvectors Corresponding the Eigenvalues	80
9	Eigenvectors for Non-dominated Sample Points	82
10	k -EMOSS Results for $k = 4$	86
11	Comparing PCA Based and Dominance Based Techniques	87
12	Convergence of NPGA and SPEA II	115
13	Probabilistic Knapsack Problem	150
14	Solutions from Pareto Layers	153
15	Technology Constraint Matrix	163
16	Responses Considered	197
17	Responses and Constraint Values for Baseline Aircraft	198
18	Technologies Considered	201
19	Dimensionality Reduction With k -EMOSS	206

LIST OF FIGURES

1	Design Process Paradigm Shift	3
2	Timely Matching of Requirements and Resources	4
3	F/A-22 and F/A-18E/F Schedule and Cost Growth	6
4	Knowledge Based Approach	6
5	Average Program RDT&E Cost Growth	9
6	Programs that Attained Technology Maturity at the Knowledge Points	10
7	Technology Life Cycle	11
8	Technology Identification Evaluation and Selection	17
9	Strategic Prioritization Process	21
10	The START Analytical Framework	22
11	Application of Various Methods in Technology Life Cycle	28
12	Technology Evaluation Framework	31
13	Greedy Solution for Knapsack Problem 5	41
14	Approximate and Exact Solutions to the example KP	48
15	DP Efficiency With Ordering of Items	52
16	One-On One-Off Technique	55
17	Prediction Profiler for the Knapsack Problem	58
18	Decision Chart for Technology Selection Methods	60
19	Geometric Interpretation of Weighting Method	68
20	Generating Pareto Frontier Using Weighting Method	69
21	Weighting Method with Non-Convex Pareto Frontier	70
22	Weighting Method with Skewed Pareto Frontier	71
23	The Concept of Pareto Optimality	75
24	Increase in Proportion of Non-Dominated Points with Dimensions . .	77
25	Eigenvalues for \mathbf{R}	80
26	Eigenvalues for Non-dominated Sample Points	82
27	Parallel Coordinate Plot for Three Item Combinations	84

28	Normal Boundary Intersection	89
29	General Outline of EA	93
30	Rank Assignment in FFGA	101
31	Non-dominated Sorting GA	103
32	NPGA Selection Operators	105
33	Strength Pareto Evolutionary Algorithm II	109
34	Convergence of SPEA II and NPGA	115
35	Comparing SPEA II and NPGA with True Pareto Solutions	116
36	Convergence of SPEA II Through Maximum Generations	118
37	SPEA II Convergence for Different Population Size	119
38	Impact of Archive Size on SPEA II Results	120
39	Comparing SPEA II with Random Search in 2 Dimensions	121
40	Facets of Uncertainty-Based Design	124
41	Robust Design and Reliability-Based Design	125
42	Mapping Technologies to the System	127
43	PDF of Beta Distribution with Different Parameter Values	129
44	Beta Distribution from Elicited Values	130
45	Monte Carlo Simulation	135
46	Sample Size Requirement for Monte Carlo Simulation	138
47	Probabilistic Analysis Methods	139
48	Change in Dominance Structure with Probabilistic Results	143
49	Joint and Marginal Probability Distribution	144
50	Notional Pareto Frontiers with Different Probability Levels	147
51	Deterministic Pareto Optimization	152
52	Probabilistic Pareto Optimization	154
53	Empirical CDFs for Deterministic Solution	155
54	Empirical CDFs for Deterministic and Probabilistic Solutions	157
55	Simple Technology Interactions	162
56	Technology Graph T	165

57	Permissible Combinations with $n = 10$ and $e = 10$	167
58	12 Interacting Technologies From Total of 29	168
59	Graph G for backtracking	171
60	Digraph for Enabling Technologies	172
61	Pareto Optimization and Selection of Technologies	178
62	Technology Data	182
63	Estimating Complexity of Combinatorial Technology Space	183
64	Enumerating Permissible Technology Combinations	185
65	Evaluating all Technology Combinations	186
66	Extracting Pareto Layers in the Objective Space	187
67	SPEA-II for Deterministic Pareto Optimization	191
68	Tradeoffs and Decision Making	194
69	Technology Graph	202
70	Technologies Present on 11 Dimensional Pareto Front	204
71	k -EMOSS Analysis for 11 Objectives	206
72	Technologies on 8 Dimensional Probabilistic Pareto Layers	209
73	Pareto Layer Data in JMP®	211
74	Identifying Pareto Layers and Sub-dimensional Pareto Fronts	212
75	Scatter Plot Matrix for 8-Dimensional Pareto Layers	214
76	Visualizing Sub-Dimensional Pareto Layers	216
77	Drift of a Technology Combination Over Pareto Layers	217
78	Three Dimensional Pareto Layers	218
79	Dendrogram for Cluster Analysis in Two Dimensions	219
80	Clusters in Two Dimensions	220
81	Screen Shot of JMP®	221
82	Scatter Plot Matrix with 75% Pareto Layer	223
83	Selected Technology Combinations from 75% Pareto Layer	224
84	Scatter Plot Matrix with Selected Technology Combinations	225
85	Parallel Co-ordinate Plot for Selected Technology Combinations	226

SUMMARY

Historically, aerospace development programs have frequently been marked by performance shortfalls, cost growth, and schedule slippage. New technologies included in systems are considered to be one of the major sources of this programmatic risk. Decisions regarding the choice of technologies to include in a design are therefore crucial for a successful development program. This problem of technology selection is a challenging exercise in multi-objective decision making. The complexity of this selection problem is compounded by the geometric growth of the combinatorial space with the number of technologies being considered and the uncertainties inherent in the knowledge of the technological attributes. These problems are not typically addressed in the selection methods employed in common practice. Consequently, a method is desired to aid the selection of technologies for complex systems design with consideration of the combinatorial complexity, multi-dimensionality, and the presence of uncertainties.

Several categories of techniques are explored to address the shortcomings of current approaches and to realize the goal of an efficient and effective combinatorial technology space exploration method. For the multi-objective decision making, *a posteriori* preference articulation is implemented. To realize this, a stochastic algorithm for Pareto optimization is formulated based on the concepts of SPEA2. Techniques to address the uncertain nature of technology impact on the system are also examined. Monte Carlo simulations using the surrogate models are used for uncertainty quantification. The concepts of graph theory are used for modeling and analyzing

compatibility constraints among technologies and assessing their impact on the technology combinatorial space. The overall decision making approach is enabled by the application of an uncertainty quantification technique under the framework of an efficient probabilistic Pareto optimization algorithm. As a result, multiple Pareto hyper-surfaces are obtained in a multi-dimensional objective space. Each hyper-surface represents a specified probability level, which in turn enables probabilistic comparison of various options. Other more traditional technology selection and scanning techniques such as the greedy algorithm, one-on one-off technique and designs of experiments are also explored. An *advisor* to recommend the best selection technique from amongst these options based on the complexity and scope of the problem is also an important contribution of this research.

Various techniques used for creating the exploration and decision making methodology are experimented on a benchmark knapsack problem. These techniques are used in a synergistic manner to formulate the Pareto Optimization and Selection of Technologies (POST) methodology. POST is implemented on an example technology exploration and selection problem for a 300 passenger commercial aircraft. This is a large problem with 29 technologies, 11 objectives and 4 constraints. Initially, the technologies and their system impacts are defined along with their uncertainties. The computational complexity is evaluated and the problem dimensionality reduced using a dominance structure preserving approach. Probabilistic Pareto optimization is implemented with the reduced dimensionality and three Pareto layers each corresponding to a predefined probability level are created. These Pareto layers are exported to a visualization and analysis environment enabled by JMP®. The technology combinations on these Pareto layers are explored using various visualization tools and one combination is selected. The main outcome of this research is a method based on a consistent analytical foundation to create a dynamic tradeoff environment in which decision makers can interactively explore and select technology combinations.

CHAPTER I

INTRODUCTION

What is design? According to Merriam-Webster dictionary, it is to create, fashion, execute or construct according to a plan. For engineering design, the final result of a design exercise is a complete physical description of the system. When the system is simple, like a bicycle, design process usually involves some historical data and application of basic principles of physics. But as the system becomes more and more complex, the design process also gets complex. It is no longer just regression based on some historical data and basic physics. The complex system now has to be divided into logical subsystems. A case in point is the design of a modern aircraft that is divided into subsystems such as propulsion, aerodynamics, structures and controls. The analysis of these subsystems has to be combined together and mathematical algorithms used to determine the size and weight of the aircraft. This is an iterative process.

Traditional aircraft design process is divided into three main phases namely: *a) conceptual design b) preliminary design and c) detailed design*. [1] Traditionally, for conceptual design, first order models are used to simulate the aircraft systems. In this phase, the requirements are examined, basic trade-offs considered and decisions regarding infusion of new technologies are made. The proposed concept is then passed on for preliminary design where higher fidelity models are used to analyze various subsystems. The basic configuration is frozen at this stage and only small design changes can be performed. Finally, detailed design is carried out where actual parts of the system are designed, decisions regarding fabrication and tooling are made and actual cost numbers come to light.

1.1 Motivation

With more emphasis on economics and product life-cycle given in recent years, conceptual design has become a crucial stage in the design process. It is highly desirable to come up with the right concept and assess feasibility and viability of the system with greater confidence in this initial design phase. This will help avoid costly design changes and major modifications during the system life-cycle. This line of thinking has led to a *paradigm shift* in the design process which is illustrated in the classic chart of Figure 1 (adapted from [2]). It depicts notional changes that occur in cost committed, design freedom and knowledge of a system throughout the design process. The solid curves represent variation along traditional design process and dotted curves represent changes along desired design strategy. It can be observed during initial phase that large amount of life cycle cost is committed to the design when there is little knowledge of the system. The design freedom rapidly decreases during the first two design phases and making major changes to the design becomes very expensive. This is where the basic design iterations occur and therefore, conceptual and preliminary design phases present the only opportunity where the designer can effectively and efficiently leverage cost and freedom.

1.1.1 Requirements and Resource Matching

Development of a system within time, cost and performance limits is the indicator of a successful program. Before an organization commits to a new product, requirements of the users and resources available have to be matched. The user requirements generally include some form of performance expected out of the system, while the resources available include the technologies that the developer has access to, to achieve the required performance, and the amount of time and money the customers are willing to commit for that performance. According to a study of industry best practices conducted by the U.S. Government Accountability Office (GAO), timely and accurate

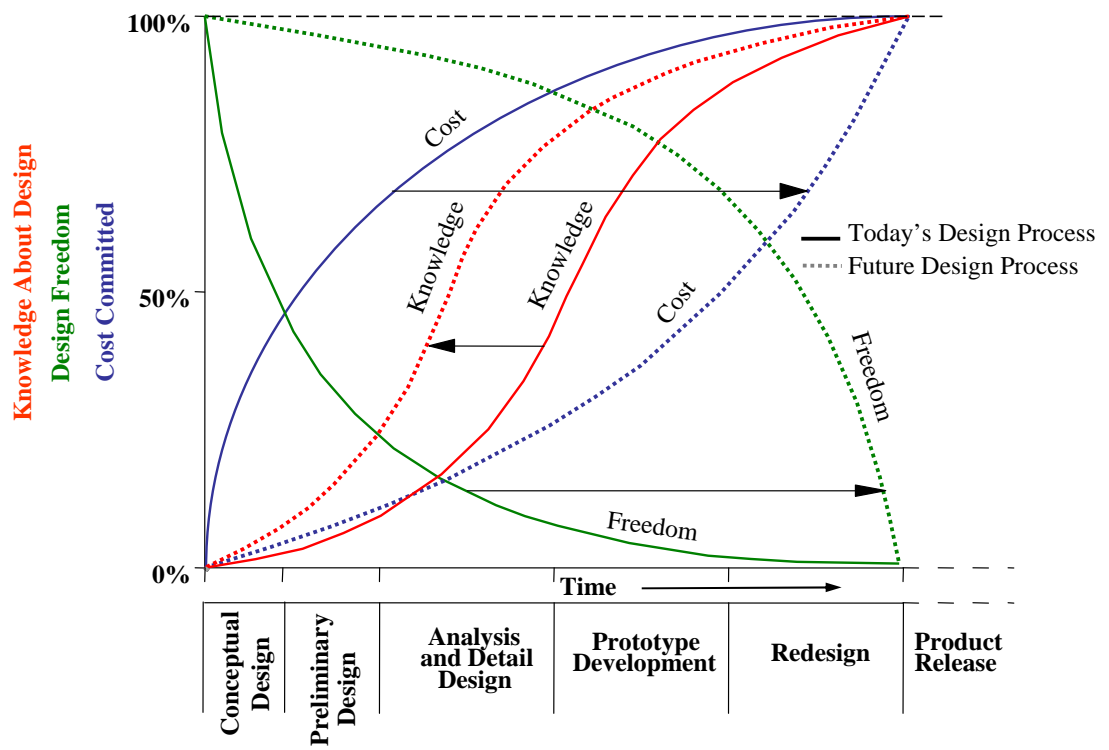


Figure 1: Design Process Paradigm Shift

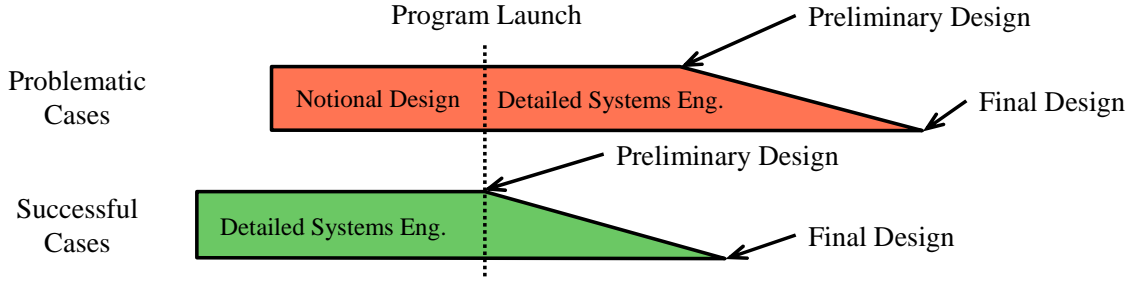


Figure 2: Timely Matching of Requirements and Resources

matching between these requirements and resources is the key differentiator between successful and unsuccessful programs [3]. In successful programs, the principles of systems engineering are used to identify areas where the customer *wants* exceed the developer resources. Some of these discrepancies are resolved by new investments the developer makes and others by investigating new technologies or alternate designs. Remaining discrepancies are resolved by relaxing the time and cost constraints involved and making required tradeoffs. This match is eventually achieved in every program, but, for successful programs, the resources are invested and program launched after this matching is achieved (Figure 2) and enough knowledge is generated about the system to complete the conceptual and preliminary design. This timely matching of requirements and resources helps delay the cost committed and relatively increase the knowledge generated in the design process as depicted in Figure 1, which in turn helps reduce overall programmatic risk.

The development programs of F/A-22 and F/A-18E/F illustrate the importance of timely and accurate matching of requirements and resources. The US Air Force and the US Navy started their respective fighter aircraft programs around the same time in the 1980's. The Air Force pursued the F/A-22 Raptor, a revolutionary aircraft with stealth and supercruise, while the Navy developed the carrier-capable F/A-18E/F Super Hornet, a new but relatively modest design based on F/A-18A/B/C/D multi-role aircraft. As depicted in Figure 3, F-22 experienced significant schedule and cost

increases as compared to F-18E/F [4]. The F-22 exceeded its original schedule for the Engineering and Manufacturing Development (EMD) phase between milestone II and III by more than 52 months while the F-18E/F was virtually on time. The cost growth for F-22 was around \$7.6 billion in 1990 dollars in contrast with the development of F-18E/F which was accomplished within initial estimates. The significant cost growth experienced by the F-22 program has resulted in the reduction of planned procurement from 648 to 183 aircrafts. A 72% decrease from the original number and in contrast with Air Force's current stated need of 381 F-22s.[5] This gap, of 198 aircrafts, between the required number and the one considered affordable by the Office of Secretary of Defense (OSD) irrefutably decreases the planned system effectiveness considerably. One of the primary reasons that contributed towards the cost and schedule growth in F-22 program, as cited by a RAND study [4] was the overly optimistic estimates for new technologies involved in the Raptor. The concurrent development of the aircraft and the technologies involved created a greater challenge for the F-22 program while the evolutionary approach adopted in F-18E/F program reduced the technical risk considerably. Thus, making accurate predictions about technology impact on the system and selecting the right mix of technologies, that will satisfy the performance and economic requirements at an early design stage, is of utmost importance for a successful program.

1.1.2 Knowledge Based Development and Acquisition

To avoid such cost and schedule growths and to deliver high quality products, leading organizations and commercial firms follow certain practices that help ensure the success of their programs. A GAO review of the practices followed by such commercial firms has shown that there are three critical points in a product development cycle where sufficient knowledge must be available to make decisions regarding large investments [6]. The first point occurs before the product development starts and a

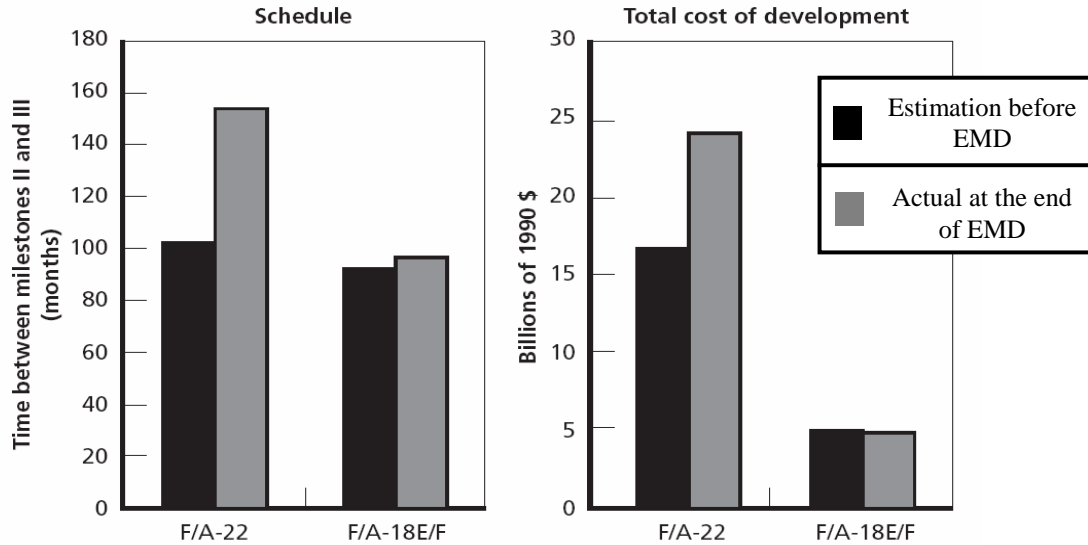


Figure 3: F/A-22 and F/A-18E/F Schedule and Cost Growth

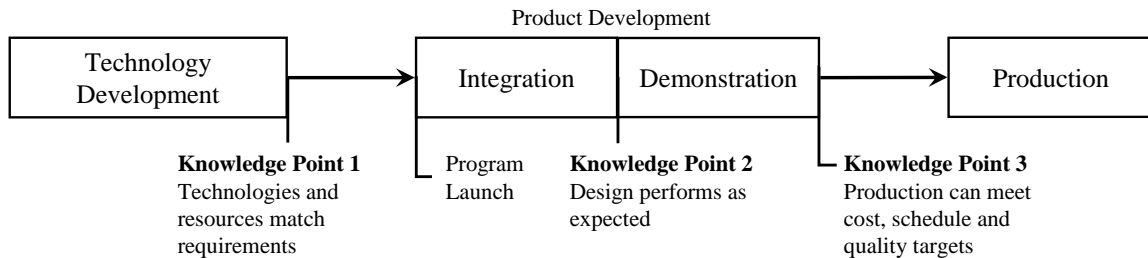


Figure 4: Knowledge Based Approach

match has been made between the customer requirements and developer resources. Second point occurs when the developer determines that the product design is stable and meets customer requirements. The third point is where the program must demonstrate that the product can be manufactured within the cost, time and quality constraints.

This practice of making important decisions after sufficient knowledge is achieved is termed as the Knowledge-Based Approach [6]. This approach is illustrated in the Figure 4 adapted from a GAO Best Practices report [7].

- *Knowledge Point 1:* This point occurs when the customer requirements and developer resources in terms of technical knowledge, time and money are matched and a sound business case is created for the product. Developers rely on historical data, systems engineering, and new technologies that are mature enough and experienced manpower to determine available resources. Gaps between needs and resources are identified and tradeoffs made at this point by communicating extensively with the customers. Emphasis at this point is to decouple technology development and product development. The program is launched only when all the technologies involved have attained sufficient maturity. Failure to do so can result in a product that costs more, take more time to develop or may not perform as expected.
- *Knowledge Point 2:* This point occurs when the developer determines that the product design is stable and will satisfy customers needs in performance and their constraints on cost and time. It generally occurs midway through the development when almost 90% of engineering drawings are completed. In case of aircrafts, the variation in design weight is a good indicator of design stability. If design stability is not achieved through the middle of product development, it may lead to expensive design changes further in the product life cycle.
- *Knowledge Point 3:* This point is reached when it is determined that the production process is mature and the product can be developed within cost, schedule and quality specification limits. Statistical process and product control tools are usually employed to determine the maturity of manufacturing process. Initiating the production before the processes are under statistical control may call for costly solutions by rework or scrap.

Table 1: Technology Readiness Levels

TRL 1	Basic principles observed and reported
TRL 2	Technology concept and/or application formulated
TRL 3	Analytical and experimental critical function and/or characteristic proof-ofconcept
TRL 4	Component and/or breadboard validation in laboratory environment
TRL 5	Component and/or breadboard validation in relevant environment
TRL 6	System/subsystem model or prototype demonstration in a relevant environment (ground or space)
TRL 7	System prototype demonstration in a space environment
TRL 8	Actual system completed and flight qualified through test and demonstration (ground or space)
TRL 9	Actual system flight proven through successful mission operations

1.1.3 Immature Technologies and their Impact

The *Knowledge Point 1* is where the decision makers have maximum leverage to affect the outcome of the program. Accurate knowledge of technologies included in the design is essential for the success of the program. For this knowledge, the technologies should be mature enough so that their impact on the system can be accurately accessed. Technology Readiness Levels (TRLs) is a systematic measurement system widely used to support assessments of technology maturity [8]. It ranks technologies on a scale of 1 through 9 based on their maturity level. Table 1 illustrates the TRL scale as described by Mankins [8] for aerospace applications. According to the GAO's best practices studies, a technology is considered mature and has low risk for starting product development if it has demonstrated its capability in the intended operational environment, i.e. it is at TRL 7 [9].

A 2006 GAO study of selected major weapons programs of the Department of Defence (DOD) found that the level of technology maturity had considerable effect on the cost growth of the program. As shown in Figure 5, the average growth in Research, Development, Test and Evaluation (RDT&E) cost for programs that started with some immature technologies ($TRL < 7$) was about 35% while the programs that began with all mature technologies ($TRL \geq 7$) experienced cost growth of only about

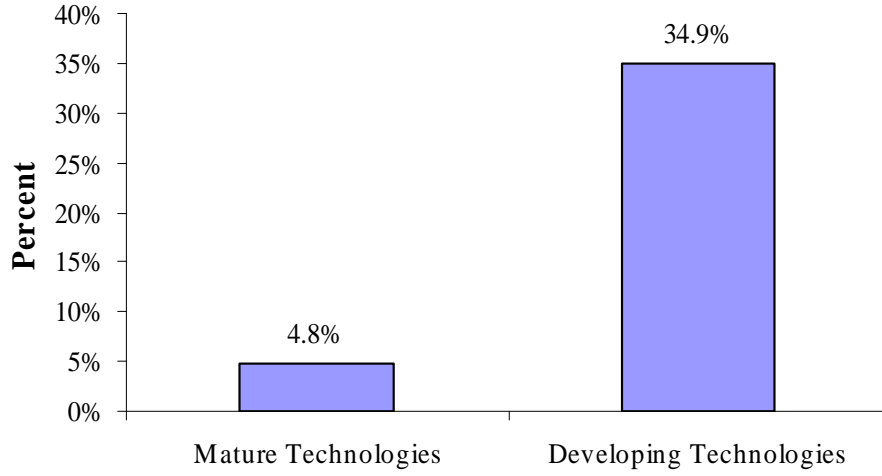


Figure 5: Average Program RDT&E Cost Growth

5% [10]. This clearly illustrates the need for achieving technology maturity before the program start.

It has been suggested that programs dealing with complex systems should move ahead only with mature technologies in them and this policy has been adopted by DoD for weapons system acquisition. However, as illustrated in Figure 6, the Knowledge Based Approach is rarely implemented [10]. Among the 2006 DoD acquisition portfolio of major weapons systems, only 10% of programs started with all mature technologies. Even at *Knowledge Point 2* and *3*, there were only 43% and 67% programs respectively, that achieved complete technology maturity. That is, even at the production decision stage there were about 33% of programs with immature technologies in them. It has been observed that decisions made on individual programs sacrifice knowledge and executability in favor of revolutionary solutions [10].

There is no doubt that the program initiated with only mature technologies will face minimum risk in terms of cost, schedule and performance. For this to occur, the design has to be evolutionary, based on proven technologies, as in case of F/A-18E/F. But given the long design cycle times of modern complex systems like a fighter aircraft, and challenging requirements involved, it is not always feasible to initiate the

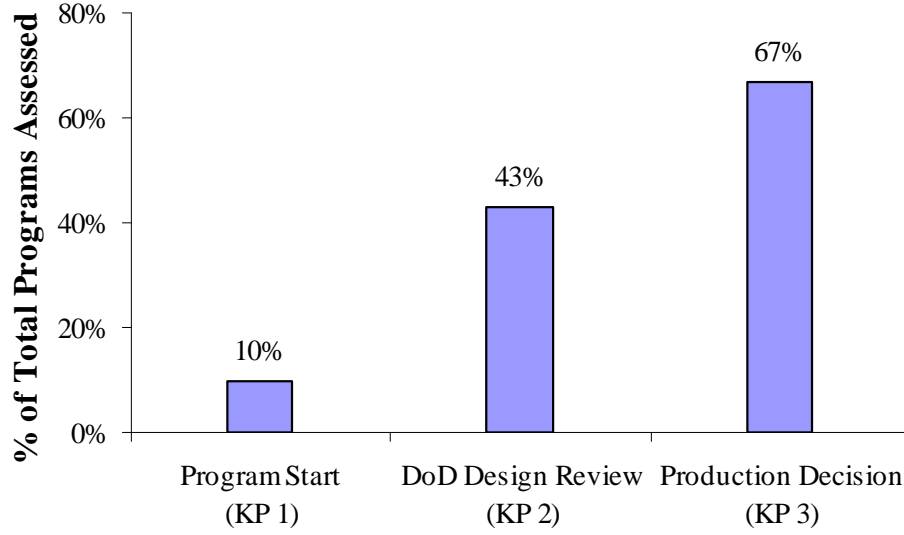


Figure 6: Programs that Attained Technology Maturity at the Knowledge Points

program with only existing technologies. Because, by the time system is operational, many of its components will become obsolete. For example, in the year 2000, F-22 had almost 600 obsolete components while the aircraft was still under development [9].

Under these conditions, technology development and product development may overlap to a certain extent as evident from Figure 6. This situation in today's world of system development arising due to challenging requirements posed by the customers provides the main motivation for this research.

1.1.4 Selecting the Right Technologies

The development of technologies leading up to their transition to a specific product is a gated process and the general flow is illustrated in the Figure 7. An important precursor for successful technology transition is good strategic planning [11]. Strategic planning can be defined as the process to identify technologies that can help achieve company's strategic goals and prioritize resources for their development. At this stage most of the technologies are in their infancy and qualitative techniques are used to select most promising of them for further development.

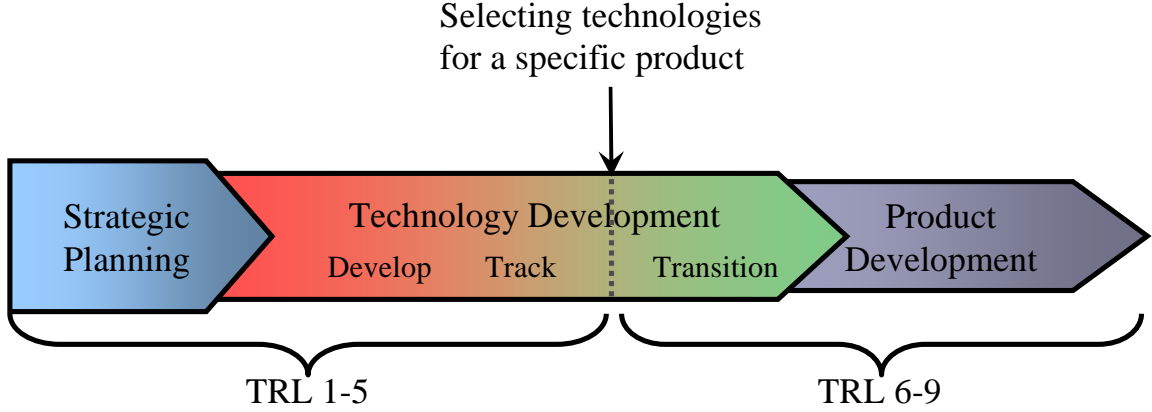


Figure 7: Technology Life Cycle

Once technologies are selected during the strategic planning, their further development is divided roughly between two different sections of the organization. The exploration and development of core technologies is the responsibility of respective research lab. The technologists develop the technologies through research and experimentation, refine the solution and can also identify the product/products that can incorporate these technologies.

When there is a requirement for a new product, in the form of a request for proposal (RFP), a product design team is assembled in the organization. Its the responsibility of this team to carry out the early conceptual designs and identify the gaps between resources and requirements. When performance or economic requirements are not met by any of the existing technologies in the design, new technologies are sought. Inputs from the technologists regarding available technologies and their maturity level are of great value at this stage. These decisions regarding selecting new technologies for a system are crucial for a successful program. Thus a decision making environment is required that helps the designers select the right group of technologies for designing a competitive system. The creation of this type of technology selection environment is the basic aim of this research.

1.2 *The Technology Selection Problem*

As described before, when a design does not meet the performance and economic requirements of the customer, new technologies have to be infused into the system. There are some important properties of the problem that have to be considered in order to create a technology selection and decision making framework. Here, the core problem is of combinatorial optimization. This has to be carried out in a multi-objective and uncertain design space. These topics are discussed in details in the following subsections.

1.2.1 Core Problem

At the heart, technology selection is a combinatorial optimization problem. Here, the best combination of technologies is to be selected, from many available, that can meet all constraints and satisfy various requirements. It is a challenging problem to solve for several reasons, one of them being the size of the combinatorial decision space. Ignoring the inter-technology constraints, like enabling and incompatibility relations, the addition of each available technology option causes a geometric increase in the size of the solution space, given by 2^n , where n is the number of technologies available. This increase is referred to as the *curse of dimensionality*. A rough idea of the computational timescales involved here can be gauged from Table 2 (adapted from [12]). As seen in the table, even if the time to evaluate all 1024 combinations of 10 technologies is a conservative 0.01 seconds, as the number of technologies goes beyond 30, the time required to evaluate the combinations becomes prohibitively large. This estimate does not even consider the computational time required to compare the combinations to find the best solutions. The number of comparisons needed can be as high as:

$$\frac{2^n}{2}(2^n - 1) \times i$$

Table 2: Time Complexity of Technology Combinatorial Space

Time Complexity	Size n					
	10	20	30	40	50	60
2^n	0.01 second	10 second	3 hours	4 months	3.5 centuries	357 millenia

where, i is the number of objectives to be tracked. As a result, an exhaustive combinatorial search becomes impractical when there is a large number of technologies.

1.2.2 Multi-Objective Design Space

The process of selecting technologies for any new complex system, such as an aircraft, is a challenging exercise in multi-objective optimization and decision making. The technologies have to be selected based on their impact on variety of objectives. Performance objectives such as range, payload, empty weight, cruise speed, specific fuel consumption (sfc), etc. have to be optimized. Apart from these performance objectives, emissions and noise variables have to be optimized in order to have a competitive aircraft. In many situations, especially during the early conceptual design phase, economic and time constraints are not known. Thus, these variables have to be considered as extra objectives to be minimized.

As the technologies are selected based on their impact on a variety of objectives rather than a single objective, the final solution is always a compromise between conflicting objectives. If optimizing the performance metrics like sfc, cruise speed, weight, etc. tend to include more and latest technologies, cost and time considerations tend to include fewer technologies that are significantly mature. Thus technology selection has to be carried out under a multi-objective decision making framework.

1.2.3 Technological Uncertainties

At the early design stages when technology decisions have to be made, the technologies themselves are not very mature as discussed previously. Thus their impact on the system under consideration cannot be estimated with high confidence. There is always

some uncertainty associated with technology impacts at this level. Moreover, at this early design stage, the system itself is not well defined and very little is known about it. Thus, even if a technology is mature, with TRL 7 for example, its impact on the system cannot be quantified with exact precision.

This uncertainty in technology impacts is propagated through the system responses in a complex manner. Thus while making technology decisions, it is imperative to consider the impact of technological uncertainties on the system responses. A probabilistic decision making process is required to accomplish this.

1.3 Research Objectives

Considering the problem stated above, the final product of this research is envisioned to be a decision making process for selecting technologies for a specific system. They are to be selected from a large pool of options that are in their early stages of development or are mature and ready for infusion. This method should be able to handle any number of technology options as long as relevant data regarding their impact on the system is available. The process has to be flexible enough to allow the decision makers or designers to select and compare various technology portfolios in real time without any significant computation involved, i.e. it should be very efficient for the decision makers. It has to be comprehensive in its consideration of all the objectives and constraints. It should be capable of accounting for uncertainties involved and should provide decision makers the capability to compare various options probabilistically. The process should involve decision makers at all critical junctures and shall be transparent, repeatable and auditable, capable of supporting electronic design reviews that are becoming a norm in the aerospace system design community. As one can imagine, technology selection problems come in variety of forms; a few technologies impacting on a single system response to the more complex ones discussed before. It is of interest to create a generic technology selection *advisor* with

techniques catering wide range of technology selection problems.

1.3.1 Research Questions

To structure this research and to facilitate the development of the aforementioned process, the following high level research questions are posed. These questions will be addressed throughout this thesis in varying details.

1. What is state of the art in technology selection process?
 - This question will lead to the previous research done in the area and give pointers for the basic framework of the solution. Important and relevant pieces from previous techniques will be identified for their use in current process. Ideas for technology selection *advisor* will also be obtained from this study.
2. How to address the multi-objective nature of the problem?
 - Search vs. Optimization? Are we interested in an optimized solution for particular objectives or a generic solution over the entire range. Various decision making methods towards this end will be discussed and the best suited for the purpose will be selected.
3. How to account for technological uncertainties while selecting technology combinations?
 - This is one of the most important question this research will try to answer. It will help select solutions based on their impact on the overall system uncertainty.

These questions will lead to many low level questions; they will be described and addressed in relevant sections of this thesis.

CHAPTER II

STATE OF ART IN TECHNOLOGY SELECTION

Before embarking on a research quest or trying to solve any problem, it is prudent to investigate previous studies done in the area. Technologies being the underlying theme of this research, major part of this chapter will review some of the past and current methods adopted for technology assessment and selection for complex systems. Traditionally, quantitative and qualitative methods are used for this purpose depending on the exact application and availability of information. For comprehensiveness, both of these types are discussed here.

The focus of this chapter is more towards studying comprehensive methodologies that deal with the process of designing a system with infusion of new technologies. The strengths and shortcomings of various methods are discussed in the light of the research goals and observations made regarding the absence, in the existing literature, of specific qualities desirable for this research. Elements of existing methods that can be used for the purpose of this research are highlighted.

2.1 Technology Identification Evaluation and Selection (TIES)

TIES is a comprehensive and structured method to allow for the design of complex systems which result in high quality and competitive cost to meet future, aggressive customer requirements. TIES brings in various techniques for technology evaluation and selection in a unified methodology that is generic enough to apply for the design of any complex system. The flow of this process is illustrated in Figure 8 and the basic theory behind this methodology has been extensively explained by Kirby [13, 14] and Mavris [15, 16].

The first step of problem definition involves mapping of customer requirements or

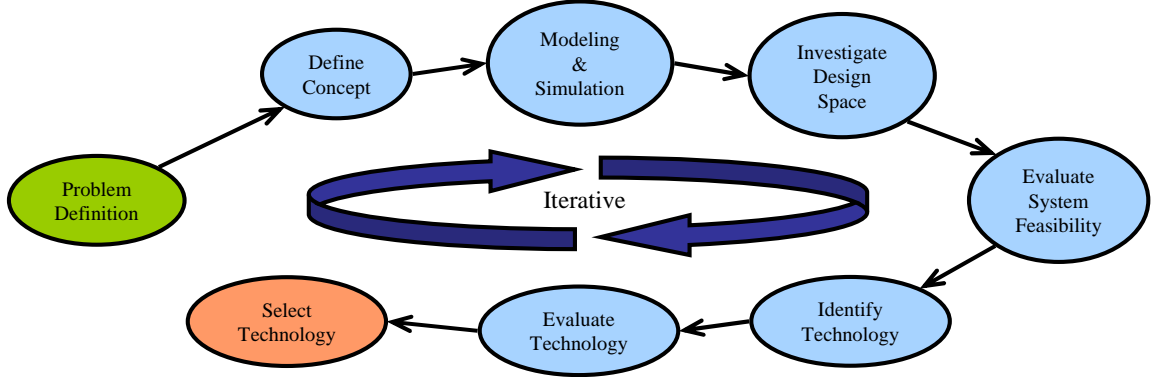


Figure 8: Technology Identification Evaluation and Selection

voice of the customer to specific design metrics or *voice of the engineer*. The voice of customers is in the form of some qualitative characteristics and this has to be translated into specific quantitative measures for the engineers and designers to work with. This is achieved through the use of brainstorming techniques such as Quality Function Deployment (QFD) [17]. This step helps establish firm system level objectives, constraints and evaluation criteria. The next step involves defining the concept space. Brainstorming using Morphological Analysis is used to accomplish this task [18]. The output of this step is a Morphological Matrix that defines the alternative design space and the definition of a baseline with which to compare different alternatives. In step three, a physics based modeling and simulation environment is created to facilitate accurate evaluation of design alternatives. The investigation of the design space is carried out in step four. For this purpose, the Response Surface Methodology (RSM) is used to bring the knowledge of high fidelity simulation codes early in the design process [19]. This step results in Cumulative Density Functions (CDFs) and Probability Density Functions (PDFs) of the objectives and the system feasibility is evaluated in step five using these CDFs and PDFs. Step five helps identify the concept *show stoppers* and the improvement required for feasibility.

To improve upon the current concept, technologies have to be identified that can

be included in the system. This is accomplished in step six and it is one of the most important steps of TIES process. The system level impact of these technologies is quantified in terms of changes in few key parameters known as technology metrics or k -factors. These technology vectors are represented in a Technology Impact Matrix (TIM). The technology under consideration may have some compatibility constraints attached to them and these are represented by Technology Compatibility Matrix (TCM). These two matrices are the primary output of this step. Once technologies are defined using TIM and TCM, each technology combination is evaluated for the system by means of Response Surface Equations (RSE). Monte Carlo simulation is used if probabilistic results are required. After the information regarding each technology combination is obtained, the best family of technology alternatives is selected using any one of the Multi Attribute Decision Making (MADM), technology frontiers or resource allocation techniques.

2.1.1 Advantages and Shortcomings

TIES is a comprehensive methodology that addresses system design problems right from the problem definition stage through technology selection. The technology evaluation model is one of the most notable feature of TIES. Quantifying technology impacts using TIM and technology incompatibilities using TCM make the evaluation model transparent and the results traceable. For early design stages when the system is not defined and technologies are immature, TIM may be the only way to capture the information.

One of the primary shortcoming of TIES methodology is in its inability to handle large number of technologies. This is because it has to evaluate each and every combination in order to apply MADM technique. As seen in the previous chapter, this number can quickly become intractable with increasing technology options. Moreover, the multi-objective and probabilistic nature of the problem is not addressed by this

method in sufficient detail.

2.2 Technology Metric Assessment and Tracking (TMAT)

The TMAT process is evolved from the combination of the High Speed Research (HSR) metrics tracking process and the TIES methodology [20, 21]. This process provides a means to optimally allocate resources to R&D tasks to meet organizations strategic goals. It is executed via five major steps:

- *Technology metric identification* is accomplished by the Integrated Product Team (IPT) using various brainstorming tools. The aim here is to identify the top level goals and their relations to specific technology metrics.
- *Technology audit scheme definition and information gathering* is meant to acquire a detailed and objective description of the technology development programs under consideration. A form of Delphi technique of self administering questionnaire is used for this purpose.
- *Technology metric assessment* is focused towards quantifying the information obtained via technology audits. This is done by defining the technology impact matrix (TIM), technology compatibility matrix (TCM) and appropriate distributions for technology uncertainty.
- *Technology metric integration* is about assessing the impact of various technologies on the organization's strategic goals. Generally a computer based modeling and simulation environment using response surface equations (RSE) is used for this purpose. Probabilistic assessment of selected solutions is accomplished using Monte Carlo simulations.
- *Technology metric sensitivity assessment* is the examination of results obtained in the previous step. The impact of each technology on the goals is visualized

in a dynamic environment. Various charts such as radar diagram, technology frontiers, etc. can be used to help decision makers arrive at an informed solution.

2.2.1 Advantages and Shortcomings

This process helps an organization track and monitor various technology development programs and allocate resources based on the strategic goals. One of the identifying feature of this process is the technique of intelligently assigning distributions to technology impacts in order to capture the associated uncertainties.

The process lacks a formal optimization framework to identify best portfolio when the number of technologies under consideration is large. Moreover, only a few solutions are analyzed probabilistically using the Monte Carlo technique.

2.3 Strategic Prioritization Process (SP2)

Developed by Kirby et al. [22], SP2 provides a structured, traceable and transparent process for planning and prioritization of various R&D programs at a strategic level for the success of any organization. Kirby and others define strategic planning as :

“a structured process through which an organization translates a vision and makes fundamental decisions that shape and guide what the organization is and what it does.”

The process is based on quality engineering methods such as QFD and Design for Six Sigma. SP2 is a five step approach as depicted in Figure 9. At the heart of this process is the link between customer requirements and technology options modeled through different interlinked decision or planning matrices. The front end of this method is a dynamic user interface that utilizes the linked matrices as its engine to perform various trade studies and prioritize the R&D programs according to decision maker’s preferences.

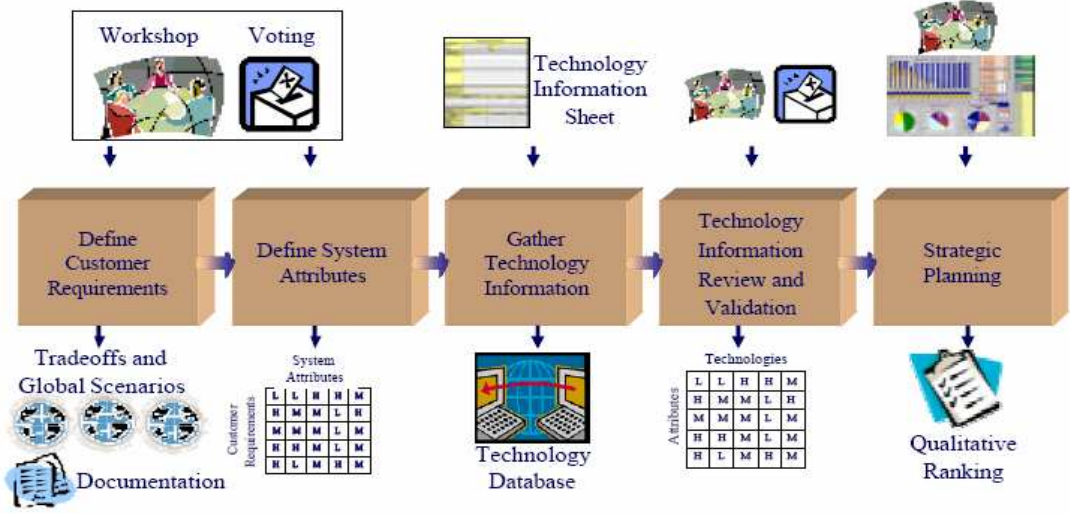


Figure 9: Strategic Prioritization Process

This method streamlines the process of defining requirements, system attributes and technologies through the participation of different levels of personnel from the organization in a series of workshops and voting exercises. This technique enables one to gather unbiased information and prevents undue influences of more powerful or vocal people. The final outcome is a prioritized list of technologies or programs to invest in for a given budget.

2.3.1 Advantages and Shortcomings

This is an excellent method tailored towards strategic planning where information is usually qualitative and a quantitative physics based approach is neither available nor preferred. The workings of the frontend of this method is fast. The customer requirements can be changed and the resultant change in the technology ranking is visualized immediately.

For the need for speed, a *Greedy algorithm* is used for ranking the technologies or programs. This is an approximate algorithm and does not give an exact answer. It cannot meaningfully handle more than one constraint. As it is designed for strategic

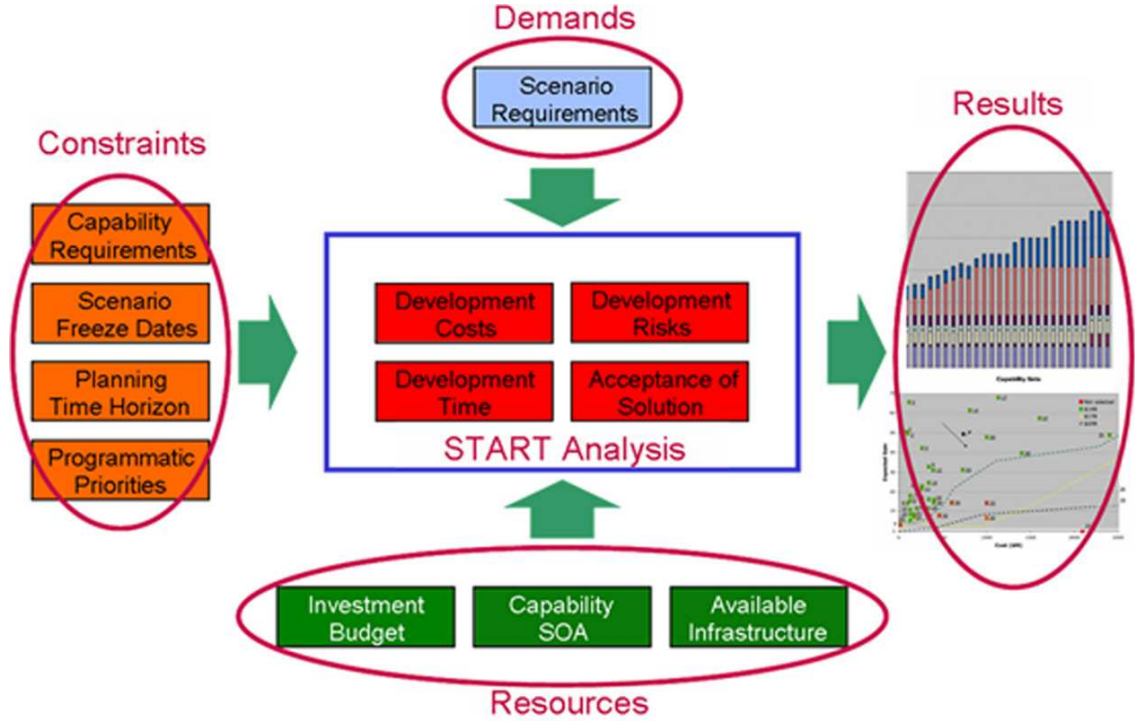


Figure 10: The START Analytical Framework

planning, SP2 does not address elements such as multi-objectivity and uncertainty that are part of the technology selection problem.

2.4 *Strategic Assessment of Risk and Technologies (START)*

The START approach has been developed at Jet Propulsion Laboratory (JPL) as a part of the drive towards addressing the NASA goals for an overall integrated agency wide approach towards systems analysis [23]. The approach provides a consistent methodological foundation for selecting and monitoring R&D tasks to enhance various NASA missions. The general procedure followed in the START process is illustrated in Figure 10 adapted from Elfes et al. [24]. START is an evolving framework and an in depth description of the current process is given by Elfes et al. [24]. Evaluation and ranking of technologies is one of the primary focus of the START approach and a few methods developed for this purpose are described below.

2.4.1 Decision Tree Assessment

A decision tree formulation is used by Manvi et al. [25] to allocate R&D resources to technology portfolio for a life detection mission to Europa. The decision tree is used to formalize the execution sequence of mission and various technologies available for each part of that mission. The figure of merit (FOM) (or probability of success) and related R&D cost are assigned to each of the options and powerful decision analysis like Monte Carlo simulations can be executed on the decision tree model. The FOM value for technologies is calculated by considering the current and required performance metric, the degree of difficulty of development and the technology readiness level of that particular technology. Hence, it has both quantitative and qualitative flavor to it. The metric for prioritization of a technology is the sensitivity of its FOM as regards to its investment divided by the initial FOM. The FOM of the system is considered to be the product of FOMs of all included technologies and the system cost is the sum of R&D cost. For optimization of the portfolio, the objective function is derived from the system FOM equation and the constraint is R&D budget available.

2.4.1.1 Advantages and Shortcomings

The primary advantage of this technique comes from the fact that it can use both quantitative and qualitative data and can be implemented in the very early stages of the design process when data regarding various system components is not readily available. Being relatively simple and easy to implement, the method lacks the rigor required while designing large scale systems. It does not address the multi-objective nature of the problems that are generally encountered while designing complex systems. The use of FOM or probability of success in the formulation does not account in detail, the uncertainty associated with technologies and their propagation on to the system performance, to be useful for robust design purposes. This method can be an excellent tool for strategic planning where the main aim is of prioritizing the

R&D projects.

2.4.2 Inference Nets

Elfes et al. [26] have used inference nets to model the relationships that link investment in technologies to mission risk and expected science return for a space mission design. They address the problem of determining the optimal technology portfolio that minimizes risk and maximizes science return of a Mars roving mission. The inference net is a graph based data flow model that allows the representation and computation of both deterministic and stochastic information. The technology uncertainty is aggregated into the mission risk by fashioning it according to the stress-strength evaluation method. A Monte-Carlo simulation is used to generate PDFs and CDFs for mission risks. This technique is extensively employed in structural analysis and is used to find the probability that an uncertain variable X is greater than another uncertain variable Y . An important component of this method is the technology development cost model that is created using historical data and expert elicitation. This allows to estimate sensitivity of mission performance with respect to R&D investment.

2.4.2.1 Advantages and Shortcomings

This method convincingly integrates the mission risk and R&D investment aspects of the design to get a clear picture of how each technology behaves in this space. The drawback of this technique is in the narrow scope of its application. The basic problem has two objectives - landing and roving, their respective risks and two technologies, one for each objective. For this problem, the stress-strength technique for accounting uncertainty is satisfactory but it is not clear how effective it would be when there are more than two objectives and technologies. The method does not address the problem of optimizing technology portfolio when there are a large number of technologies available.

2.4.3 Post-optimality Analysis

Given the uncertainties involved with any input data, it is imperative for the decision makers to know the robustness of the solution. To address this, probabilistic analysis is carried out by Adumitroaie et al. [27] on an optimal portfolio identified for a given investment budget. Two techniques were employed to investigate the robustness of the solution. The first is a parametric screening method where the value of cost and utility of each R&D task is changed incrementally and independently to determine its impact on the optimum research portfolio. This approach shows the range of cost and utility over which the portfolio remains constant. The next technique employed is a Monte Carlo simulation where the cost and utility of each task is varied simultaneously and the portfolio is optimized each time. The simulation is carried out for 1000 runs and the result is in form of the frequency of occurrence of each task in the optimized portfolio of those runs. The results from these two techniques are combined into a single chart on cost and utility axis where the individual R&D tasks are designated as *robustly selected*, *robustly rejected* or *trade candidates*. The authors also have performed k -best analysis on the optimum portfolio. This analysis suggests k suboptimal portfolios that are closest to the optimal one for the given budget level. This concept helps decision makers to take into account factors that cannot be modeled quantitatively while selecting the portfolio.

2.4.3.1 Advantages and Shortcomings

The probabilistic analysis performed using two techniques gives good depth to the results obtained after optimization. The technique is efficient for strategic planning purposes where information regarding the system and R&D tasks is limited, hence the use of only two dimensional space of utility and cost. The k -best analysis can prove extremely useful while considering recourse actions or backup plans. The main limitation of this technique is that it is carried out post-optimality. Thus all options

are not compared probabilistically to reach the optimal solution. Moreover, much information is lost because of collapsing the performance objectives into a single utility factor. This may not be appropriate while designing complex systems such as an aircraft.

2.5 Other Techniques for Technology Assessment and Portfolio Planning

Apart from the aforementioned methods, there is a plethora of literature available on technology assessment and portfolio planning; a few of them are briefly described in the following sections.

2.5.1 Strategic Technology Assessment

Shishko et al. [28] examine the use of real options valuation for assessing technologies in the context of prioritizing NASA technology portfolio for given investment. Here, technology developments are treated as assets with uncertain payoffs that may result in significant returns with limited losses. This technique enables NASA to decide whether to invest or not in a mission that uses those technology options, and also gives them flexibility of choosing when to invest or change the mission. R&D Project Portfolio Matrix is used by Mikkola [29] as a tool for analyzing R&D portfolios by linking competitive advantages of the organization to the customer benefits provided by the projects. It is a graphical technique that facilitates the selection of projects with the highest potential of success. Wyk [30] proposed the use of strategic technology scanning as a means to strengthen the link between technology and corporate strategy. Wyk states a few requirements for the scanning activity such as: its results should be directly useful for strategic planning process and it should contribute to the technology foresight of the managers. Incidentally, the aforementioned SP2 process by Kirby et al. [22] fulfil these requirements and can be considered as a procedure

for strategic technology scanning. A Cross-impact Hierarchy Process (CHP) is proposed by Cho and Kwon [31] to assist in ranking of a large number of interdependent technology alternatives. Here, an Analytic Hierarchy Process (AHP), which is used extensively for R&D project selection, is linked with Cross-Impact Analysis which models the interactions among various R&D projects.

There exist many techniques for analyzing technologies and their consequences; all these methods fit into the field of study known as Technology Futures Analysis (TFA). Porter et al. [32] have an excellent compilation of existing methods for TFA and provide some valuable insights into this field. Even though TFA is focused primarily towards strategic decision making, be it at a corporation level, national level or global level, there are many techniques that can be adopted for the problem at hand.

2.5.2 Portfolio Planning

Based on the TIES formulation, Utturwar et al. [33] devised a two step optimization process for technology selection. In the first step, a gradient based optimizer is used to obtain a vector (k_{opt}) of optimal k -factors¹ for the desired response. In the second step, a combinatorial optimization is used in the discrete space to obtain optimum technology combination that produces a k -vector closest to k_{opt} . A Pareto Ant Colony Optimization is introduced by Doerner et al. [34] as an approach for research portfolio selection in a multi-objective space. Ant Colony Optimization is also used by Villeneuve [35] for exploration and selection of concepts and technologies for aerospace architectures. Sun and Ma [36] have used the packing-multiple-boxes (or multi-knapsack) model to select and schedule candidate R&D projects. This method attempts to maximize the total value of selected R&D tasks concurrently trying to schedule the starting time of each task so that the total cost is within the

¹ k -factors as mentioned in the TIES section

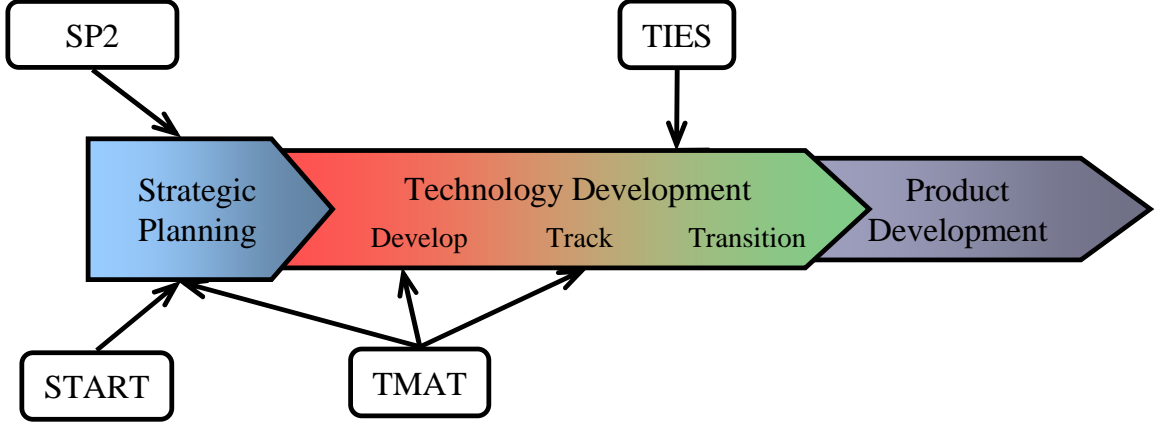


Figure 11: Application of Various Methods in Technology Life Cycle

allocated budget of each year. A hybrid evolutionary approach has been implemented by Subbu et al. [37] for financial portfolio optimization. The portfolio planning for financial assets and R&D technologies is almost similar, only difference being the evaluation of assets and technologies. This approach uses evolutionary computation with linear programming to identify the efficient frontier in the risk vs. return space and is currently used in the financial decision making industry.

2.6 Observations

Previous sections have described a few techniques that are relevant to this research. These techniques and methods are applicable for decision making and management at different stages of technology development life cycle as illustrated in Figure 11. This pairing of methods with development stages is the author's own opinion based on the application examples in the respective literature. Though, it is understood that some of them can be modified according to the requirements, for example, START is primarily developed for strategic decision making but can be extended for selecting technologies for a specific product.

In order to qualitatively compare various methods, three main criteria are identified that are congruous with the research objectives. These are:

Table 3: Method Comparison

	No. of Tech	Tradeoffs	Uncertainty Assessment
TIES	3	2	2
START	1	3	2
SP2	2	3	3
TMAT	3	2	2

Table 4: Legend for Method Comparison

	No. of Tech	Tradeoffs	Uncertainty Assessment
Best (1)	Any	Interactive	Pre-optimality
Better (2)	Large	<i>A Posteriori</i>	Post-optimality
Good (3)	Limited	<i>A Priori</i>	None

- *Number of technologies* that can be efficiently handled by the method. More technologies, more combinations to evaluate and greater computational time. It is desirable to have a method that can work with large number of technologies.
- *Effectiveness of tradeoffs* in a multi-dimensional objective space. There are various ways and stages in the process where tradeoffs can be made in the multi-dimensional design space. Decision makers prefer making tradeoffs when they are aware of the entire design space and all the options available.
- *Uncertainty assessment* technique employed by the method. It is preferable to assess uncertainty involved in all the options available and then select the best solution based on its probability level.

An objective comparison of the most relevant techniques based on the aforementioned criteria is presented in Table 3 with legend in Table 4.

As evident from the table, only START can handle a large number of technologies at a time. It uses a Knapsack algorithm that can optimize from a large number of available technology options. But, when coupled with risk assessment the computational time can be significant. TIES and others compute all the combinations, which can be very large for a large number of technologies and hence computationally

infeasible.

Most of the methods use a form of utility function or overall evaluation criteria (OEC) for making tradeoffs. While they simplify the optimization process significantly, the decision makers have to make tradeoffs in the absence of complete knowledge of the design space. An ideal approach would be to make tradeoffs as the optimization goes on and steer the process towards the preferred section of design space. This, in most cases is not advisable as the optimization process for a complex system may take several hours or days and the approach becomes very inefficient for the decision makers.

Current methods that employ uncertainty assessment execute it post-optimally, i.e. they select a few good deterministic solutions, apply uncertainties to the inputs and obtain cumulative distribution function (CDF) or probability density function (PDF) of the objectives, and in almost all cases, using Monte Carlo simulations. As mentioned before, it is preferable to compare different solutions and optimize by considering uncertainty right from the beginning, i.e. a form of *pre-optimality* uncertainty assessment.

2.6.1 Useful Techniques

Based on the above discussion, some important techniques addressing particular aspects of the problem come to light that can be used to satisfy the research goals of this thesis. Most of these are based on the TIES methodology.

The TIES and TMAP methods provide an excellent *technology evaluation framework* that can be used for the current research. This is notionally illustrated in Figure 12. The fundamental premise of this approach is that system level impact of most technologies can be quantified in terms of few key parameters known as technology metrics or *k*-factors. The most important *k*-factors for the system are identified and functionally related to the system responses through system models

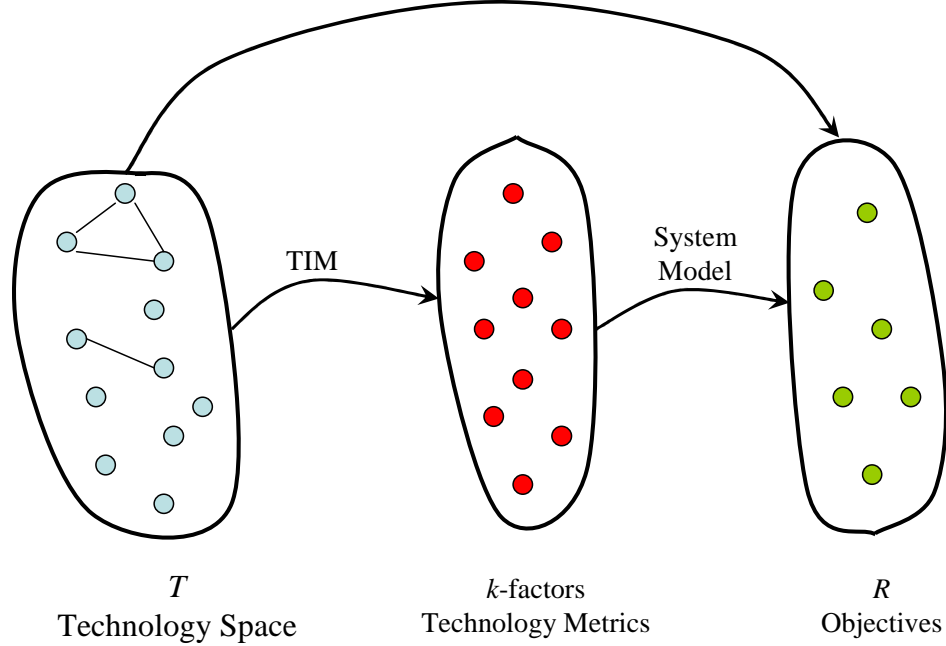


Figure 12: Technology Evaluation Framework

or surrogate models. The technologies are mapped to these k -factors by estimating their impact on them. This mapping is formalized in a Technology Impact Matrix (TIM). Thus, an accurate estimate of technology's impact on the system is obtained. TIES also provide a technique to formalize incompatibilities among technologies via the Technology Compatibility Matrix (TCM). It is important to account for such constraints among technologies as their existence changes the combinatorial design space. TIES also implements the Response Surface Methodology (RSM) for fast and accurate evaluation of system responses.

The above technology evaluation technique is used for the purpose of this thesis. Response surface equations obtained via RSM will be used for mapping k -factors and the system responses.

2.7 Summary

The review of existing literature on technology assessment and selection has shown that there is no comprehensive method that can handle a large number of technology

options and at the same time account for technological uncertainty right from the start of the process. When a large number of combinations are evaluated, the tradeoffs are attempted without the knowledge of the entire design space. Even though there is no single method that can satisfy the research goals, a framework for technology evaluation from the TIES methodology has been identified and this will be used as a foundation on which a novel approach for technology selection will be built.

It was also observed from the literature that authors have used various types of algorithms, from greedy to knapsack algorithm, for technology optimization. It should be interesting to investigate these algorithms that can help create a technology selection *advisor* as mentioned in the previous chapter.

From the literature review, there are four major themes that come to the forefront of this research.

- *Algorithms for Technology Selection* to study various algorithms available for combinatorial optimization. This will help create an *advisor* for solving wide range of technology problems.
- *Multi Objective Decision Making (MODM)* looks into the question of search vs. optimization with the aim of providing an efficient and effective tradeoff environment to the decision makers and managers.
- *Uncertainties and Probabilistic Evaluation* is for ways to account for technological uncertainties and make decisions based on probabilistic evaluation of various options.
- *Technology Compatibility Constraints* is about modeling and analyzing the compatibility constraints with the aim of assessing their impact on the design space.

Each of these themes are addressed in varying details in the following chapters of this thesis.

CHAPTER III

ALGORITHMS FOR TECHNOLOGY SELECTION

As noted in the previous chapter, there are a few existing methodologies catered towards technology selection. These methods use some fundamental algorithms and techniques to actually select the best combination of technologies from the available options. This chapter will explain the inner workings of these algorithms. Some statistics based techniques to investigate the overall technology combinatorial space are also described.

The technology selection problem being a combinatorial problem is similar in structure to the Knapsack Problem (KP). This problem is introduced in the first section and is chosen as a benchmark problem to demonstrate various algorithms and techniques. The algorithms are categorized in two main families: approximate and exact algorithms. Two examples for each families are discussed in detail. Other investigative techniques are also discussed. Finally, a framework for technology selection advisor based on the algorithms and techniques discussed is provided.

3.1 Technology Selection and the Knapsack Problem

For the combinatorial optimization of technologies, there are requirements involved and constraints to be satisfied. In other words, we have to fill a bag with technologies such that the collection meets certain objectives and satisfies the constraints. Viewing the problem with this perspective, it is analogous to the *Knapsack Problem* studied in the field of theoretical computation and mathematics.

The Knapsack Problem (KP) is a well known combinatorial optimization problem. Here, given a set of items with known values and weights, one has to pack the knapsack with a subset of items, such that the sum of weights of the selected items does not

exceed the capacity of the knapsack, and the sum of the values of the selected items is maximal. When there is only a single unit of each item in the set that can either be included in the bag or left out, the problem is known as 0-1 KP. This optimization problem is formally defined as:

Given a set S of n items and a knapsack with,

v_i = value of item i ,

w_i = weight of item i ,

W = capacity of the knapsack,

select a subset of the items so as to

$$\text{maximize} \quad V = \sum_{i=1}^n v_i x_i \quad (1)$$

$$\text{subject to} \quad \sum_{i=1}^n w_i x_i \leq W \quad (2)$$

where,

$$x_i = \begin{cases} 1 & \text{if item } i \text{ is selected;} \\ 0 & \text{otherwise.} \end{cases} \quad i \in N = \{1, 2, \dots, n\}$$

The optimization problem shown above is NP-hard (Non-deterministic Polynomial time) and when it is constructed as a decision problem, it is an NP-complete problem¹.

The formal definition of the knapsack decision problem is as follows:

Instance: A set S of n items. Each item i has value v_i and weight w_i (v_i and w_i may be scalar or vectors). A limit W for weight and V for value.

Question: Is there a subset $K \subseteq S$ such that the sum of the weights of items in K is at most W and the sum of values of items in K is at least V .

¹An extensive overview of the theory of complexity and NP-completeness is beyond the scope of this thesis; a comprehensive description of the theory, concepts and many NP-complete problems is provided by Garey and Johnson.[12]

There are many variations of the KP that are intensively studied such as *Multiple-Choice Knapsack Problem*, *Bounded* and *Unbounded Knapsack Problem*, *Subset-Sum Problem*, *0-1 Multiple Knapsack Problem*, etc. Martello and Toth[38] and Pisinger[39] provide excellent theoretical explanation of the KPs along with many exact and approximate algorithms used for solving them.

The set S of n items in the KP above can be compared to the set T of t technologies of technology selection problem. This problem has many objectives in contrast to the KP that only considers the value and the weight. The other significant difference from KP is that technologies interact within themselves and with the system in a very complex ways, all of which must be accounted for in the technology evaluation model if the results are to be useful. Even though there are major differences between two problems, the core is quite similar. The technology selection problem can be considered as a generalization of the KP problem. In other words, it can be reduced to KP and proved to be NP-hard. This means that the problem at hand is extremely difficult and intractable.

The knowledge that the problem is NP-hard and similar to KP provide valuable information regarding the direction of appropriate approach and types of algorithms that can be used. It hints towards the fact that an exact algorithm may not be feasible for large technology problems.

3.1.1 Benchmark Knapsack Problem

Most of the algorithms used for technology selection problems have been rigorously studies to solve the Knapsack and other NP-hard problems. Considering the similarity of technology selection problem with 0-1 Knapsack Problem (KP), a multi-objective and multi-constraint KP is devised as a benchmark problem to demonstrate and compare different algorithms. This KP has 16 items to choose from. Table 5 describes the problem where each item has three *values* and two *weights* assigned to it. A

Table 5: Example Knapsack Problem

Item No.	Value			Weight	
	V_1	V_2	V_3	W_1	W_2
1	7	9	5	6	3
2	3	6	4	7	10
3	2	7	9	10	1
4	6	5	5	9	4
5	1	5	1	1	4
6	5	4	7	6	5
7	10	4	6	10	2
8	3	5	2	5	5
9	8	4	10	7	3
10	4	10	5	6	7
11	3	7	6	6	6
12	10	6	9	5	2
13	7	9	2	4	10
14	7	4	10	5	7
15	5	10	8	3	6
16	8	8	8	8	8
Constraint on weight				40	30

combination of items has to be selected that maximizes the overall value while being within the weight constraint of 40 and 30 respectively.

3.2 *Approximate Algorithms*

As the technology selection problem is NP-hard, some instances of the problem may not be optimally solved within the stipulated time period. In such situations, approximate algorithms or heuristics are a viable option to search for near-optimal solutions. Moreover, large scale technology selection problems seldom require exact optimal solutions and good, feasible solutions are equally valuable. When an algorithm produces results that are within a *guaranteed* range of the optimal value, it is called an approximate algorithm. Heuristics, on the other hand are algorithms with no guarantee on either the degree of approximation or the running time [40].

There is considerable amount of literature available on such algorithms for KP and similar NP-hard problems with Martello and Toth [38], and Ibaraki [41] being

some of the most comprehensive resources. Some of the approximate algorithms and heuristics described by Ibaraki [41] are:

- Greedy methods
- Stingy methods
- Random search - Monte Carlo methods
- Relaxation methods
- Partitioning methods
- Partial enumeration or space reduction methods
- Iterative improvement methods - Tabu search, Evolutionary algorithms
- Simulated annealing methods

Theoretically, techniques based on partial enumeration are considered superior as they tend to exploit the structural properties of the problem, as compared to random search or simulated annealing. Any enumeration based exact algorithm such as branch-and-bound or dynamic programming (described later in the chapter) can be converted into an approximation scheme based on partial enumeration by considering a stopping criterion. Ibaraki [41] suggests stopping criteria based on relative error and number of nodes visited by the algorithm.

Greedy algorithm, a rather simple approximation scheme and Monte-Carlo or random search methods, one of the earliest form of heuristics are described in the following subsections.

3.2.1 Greedy Algorithms

A greedy strategy finds an optimal solution by making a series of decisions. These decisions, made at each stage, are the best choice at that moment. The problems

that can be solved by greedy strategy have two distinguishing characteristics: the *greedy choice property* and *optimal substructure* [42]. A problem is said to possess greedy choice property when a globally optimal solution can be reached by making locally optimal choices. The decision made in greedy strategy depends on the choices made in the previous stages, but not considering future choices. Moreover greedy strategy does not recommend revisiting a decision, as opposed to other mathematical programming techniques. This strategy usually works in top-down fashion, reducing the problem iteratively by making greedy choice at each stage. A problem is said to have an optimal substructure if the optimal solution to the problem consists of optimal solutions to its subproblems.

The 0-1 KP does exhibit optimal substructure property but does not have the greedy choice property. But, greedy algorithm exploits the greedy choice property of the, closely related, *Fractional or Continuous Knapsack Problem (CKP)* to determine an approximate solution to 0-1 KP. CKP is the *linear programming relaxation* of the 0-1 KP. It is the most natural, and historically the first relaxation of the 0-1 KP [38] and obtained by removing the integrality constraint on the items x_i :

$$\text{maximize} \quad \sum_{i=1}^n v_i x_i \quad (3)$$

$$\text{subject to} \quad \sum_{i=1}^n w_i x_i \leq W \quad (4)$$

$$0 \leq x_i \leq 1, \quad i \in N = \{1, 2, \dots, n\} \quad (5)$$

assuming for simplicity,

$$v_i, w_i, W \in \mathbb{Z}^+$$

$$\sum_{i=1}^n w_i > W$$

$$w_i \leq W, \quad i \in N$$

A classical solution to this problem is demonstrated by Dantzig [43] in a graphical

manner. In mathematical terms, it goes by sorting the items in the following order:

$$\frac{v_1}{w_1} \geq \frac{v_2}{w_2} \geq \dots \geq \frac{v_i}{w_i} \quad (6)$$

Then, each item is consecutively added to the knapsack until the first item c is found that does not fit. This is called the *critical item*. This approach leads to the optimal solution to the CKP and is formally stated as:

$$x_i = \begin{cases} 1 & \text{for } i = 1, \dots, c-1, \\ 0 & \text{for } i = c+1, \dots, n, \end{cases}$$

$$x_c = \frac{\overline{W}}{w_c}, \quad \text{where, } \overline{W} = W - \sum_{i=1}^{c-1} w_i$$

and the value of optimal solution is,

$$V(CKP) = \sum_{i=1}^{c-1} v_i + \overline{W} \frac{v_c}{w_c} \quad (7)$$

Two notable facts emerge from this solution. First, the optimal solution x is maximal, that is $\sum_{i=1}^n w_i x_i = W$. The other fact is that all items are either included ($x_i = 1$) or not included ($x_i = 0$) in the solution except only one item, item c , which has a fractional value ($0 \leq x_c \leq 1$). This second fact is exploited by the greedy algorithm for approximately solving 0-1 KP. Setting x_c as zero gives a feasible solution to the 0-1 KP. The value of this solution is $V' = \sum_{i=1}^{c-1} v_i$. It can be assumed that for most problem instances, V' is quite close to the optimal value V , which is bounded as $V' \leq V \leq (V' + v_c)$. However, the worst case performance ratio V'/V can be very bad as shown by the following example. Consider a problem instance with $n = 2$, $v_1 = w_1 = 1$, $v_2 = w_2 = k$ and $W = k$ for which $V' = 1$ and $V = k$ [38]. The performance ratios can be close to zero for $k \rightarrow \infty$. This performance ratio can be improved by considering a feasible solution given by only the critical item. Hence, $\hat{V} = \max(V', v_c)$. This changes the bounds on V as $\hat{V} \leq V \leq 2\hat{V}$. Thus, the worst case performance ratio, \hat{V}/V , for the new formulation is $1/2$.

The most popular approach for greedy algorithm is to order the items based on Equation 6 and add the items according to increasing indices till the knapsack is full. Here, items 1 through $c - 1$ are always included and any item, thereafter, that can fit in the remaining space is added to the knapsack. The worst case performance is improved to $1/2$ by also considering the solution with maximum value item alone. Algorithm 1 describes a pseudocode for the greedy algorithm used to solve 0-1 KP. The time complexity for initial sorting is $\mathcal{O}(n \log n)$, adding $\mathcal{O}(n)$ for the complete algorithm.

Algorithm 1 Greedy Algorithm

Require: Items **Sorted** according to Equation 6

```

1: procedure GREEDYKP( $n, v, w, W$ )
2:    $x \Leftarrow 0$ 
3:    $\overline{W} \Leftarrow W$ 
4:    $V \Leftarrow 0$ 
5:   for  $i \Leftarrow 1, n$  do
6:     if  $\overline{W} \geq w_i$  then
7:        $x_i \Leftarrow 1$ 
8:        $\overline{W} \Leftarrow \overline{W} - w_i$ 
9:        $V \Leftarrow V + v_i$ 
10:    end if
11:  end for
12:   $[\hat{i}, \hat{v}] \Leftarrow \text{maximum}(v)$    $\triangleright$  Here,  $\hat{i}$  is the index of item with maximum value  $\hat{v}$ 
13:  if  $\hat{v} \geq V$  then
14:     $x \Leftarrow 0$ 
15:     $x_{\hat{i}} \Leftarrow 1$ 
16:     $V \Leftarrow \hat{v}$ 
17:  end if
18:  return  $x, V$ 
19: end procedure

```

The KP in Table 5 is solved approximately using Algorithm 1. As this algorithm is designed to consider only one value, V_1, V_2 and V_3 are merged into a single utility function given by Equation 8.

$$V = V_1 + V_2 + V_3 \tag{8}$$

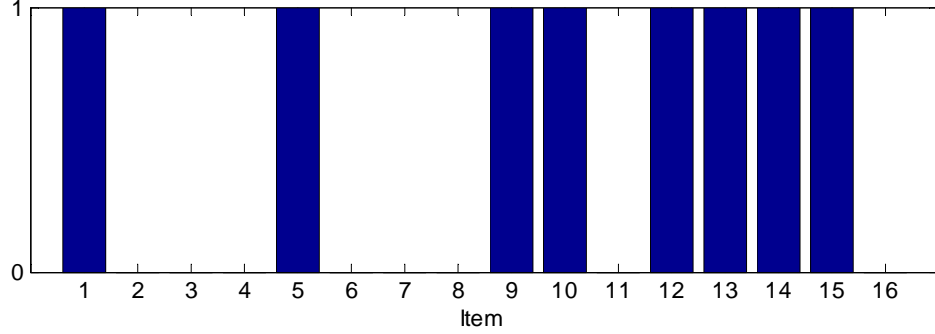


Figure 13: Greedy Solution for Knapsack Problem 5

For the weight constraint, only W_1 is considered. The approximate solution is:

$$x = [1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0]$$

$$V = 49 + 57 + 50 = 156$$

$$\hat{W}_1 = 37$$

The solution is graphically illustrated in Figure 13 with the horizontal axis showing the item number.

3.2.1.1 Advantages and Shortcomings

The main advantage of using greedy algorithm for technology selection problem is that it is extremely fast when compared to other techniques, especially when dealing with large number of options. This advantage is exploited by Kirby and others [22] in the final step of SP2 process, where a greedy algorithm is used for resource constrained program prioritization task in real time. The other advantage of this, and also other approximate algorithms, is that the solution lies within a proven bound around the optimal value. Thus, even though greedy is an approximate algorithm, there is some degree of certainty to its solutions. Apart from these advantages, greedy algorithms are very simple to implement.

One of the major drawbacks of greedy algorithms is that they are approximate in nature. They cannot be used for problems where exact solution is required.

The only way one can account for multiple objectives is using some form of utility function. This may not be an ideal approach for technology selection problems as explained in the following chapters. Moreover, this algorithm can only handle one constraint.

3.2.2 Monte-Carlo Methods

Monte-Carlo methods, also known as random search methods are some of the simplest probabilistic search methods. It consists of uniformly sampling random points from the combinatorial design space and retaining the best point that also satisfies the constraints. Considering a simple technology selection or knapsack problem with n items, the optima is from 2^n combinations. Assuming that there is only one optimal point in the combinatorial space, the probability of a randomly selected point being optimal is $1/2^n$. The probability of optimum not being found after k trials is $(1 - \frac{1}{2^n})^k$ and the probability of success is:

$$S = 1 - (1 - \frac{1}{2^n})^k \quad (9)$$

Solving for k ,

$$k = \frac{\ln(1 - S)}{\ln(1 - \frac{1}{2^n})} \quad (10)$$

The Equation 10 defines the number of trials required for a given problem and desired probability of success.

Considering the KP of Table 5, there are 16 items to consider. Thus probability of any one combination being optimum is $1/65,536$. Now, solving for number of trials required for 90% success rate, we get $k = 150,900$ which is more than double the total number of combinations in the design space. And, if one wants to be 99% certain that they have reached the optimum, 301,800 trials are required. Moreover, if only 65,536 random trials are performed, the probability of achieving optimum value is only about 63%. From this perspective, Monte-Carlo methods are clearly undesirable. But, lets consider a scenario where there are 10 points including the optimal, that

are good enough and will suffice for our requirements. Now, the probability of any one random combination being a satisfactory solution is $10/65,536$. For 90 and 99% certainty, the number of trials required are about 15,000 and 30,000 respectively. This number is significantly lower than the total number of combinations and this form of interpretation is what gives Monte-Carlo methods its strength. A pseudocode for Monte-Carlo search is illustrated in Algorithm 2.

Algorithm 2 Monte-Carlo Search

```

1: procedure MCKP( $n, v, w, W, t$ )
2:    $V \leftarrow 0$ 
3:   for  $i \leftarrow 1, t$  do
4:      $\bar{x} \leftarrow$  binary random array of size  $(1, n)$ 
5:      $\bar{v} \leftarrow$  value for  $\bar{x}$ 
6:      $\bar{w} \leftarrow$  weight for  $\bar{x}$ 
7:     if all  $\bar{w} \leq W$  and  $\bar{v} > V$  then
8:        $V \leftarrow \bar{v}$ 
9:        $x \leftarrow \bar{x}$ 
10:    end if
11:  end for
12:  return  $x, V$ 
13: end procedure

```

Running a 15,000 trial Monte-Carlo search on the KP with objective of increasing the sum of *values* and considering only W_1 as constrain, we get the following solution:

$$x = [1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1]$$

$$V = 49 + 61 + 48 = 158$$

$$\hat{W}_1 = 38$$

We can be 90% certain that this is one of the 10 best solutions. This solution is little better than the greedy solution of the previous section. This technique can easily be adopted for multiple constraints as shown in Algorithm 2. The following solution is

obtained after 15,000 trials with W_1 and W_2 as constraints:

$$x = [1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0]$$

$$V = 45 + 48 + 44 = 137$$

$$\hat{W}_1 = 38$$

$$\hat{W}_2 = 27$$

Again, we can be 90% certain that this is one of the 10 best solutions when considering both constraints.

3.2.2.1 Advantages and Shortcomings

The simplicity to implement is the main advantage of this technique. As demonstrated, it is extremely easy to consider multiple constraints with this method. When each trial is computationally cheap, as for the example KP here or some elementary technology selection problem, the number of trials can be increased significantly to increase the confidence in results.

The same property can be considered as a drawback when each trial is computationally expensive, as in many instances of technology selection problems, and one has to find the best solution with minimum number of iterations. Though there are non-domination based techniques², the most straight forward way of considering multiple objectives is via merging them into a single objective. This may not be an ideal solution for some applications. Moreover, as this method is based on randomness, there is a possibility, albeit minuscule, that the solution offered is randomly bad.

3.3 Exact Algorithms

There are situations, when feasibility and viability of the design are at stake, that an exact solution to the technology selection problem is required. Exact algorithms

²More on this in Chapter 4.

for the 0-1 KP problems rely on enumeration. For most instances of these problems, complete enumeration is seldom required and it is possible to exploit the underlying structure of the problem to design an efficient enumerative algorithm. Problems with moderate number of technology options can be solved in short time even if the computational complexity for such algorithms is exponential.

Some of the prominent enumerative algorithms for solving KP are described by Martello and Toth [38]. Balas and Zemel [44] describe an algorithm to solve 0-1 KP based on the concept of the *core problem*. They documented via various experiments that the solution to the linear relaxation of 0-1 KP is very close to the exact 0-1 KP solution. Only a few variables needed to be changed around the *critical item* in order to obtain the integer solution for 0-1 KP. This problem is denoted as the *core problem* associated with the 0-1 KP. Efficient exact algorithms based on this concept are proposed by Pisinger [39].

Almost all of the exact algorithms are based on two primary enumerative techniques: *branch and bound* and *dynamic programming*. These are explained in the following subsections.

3.3.1 Branch and Bound Algorithms

Branch and bound algorithms are exponential in time for the worst case scenario but can be intelligently designed to work efficiently for typical problem instances. This method conducts the search on a tree of all feasible solutions and reaches the optimum by solving the subproblems along the way. At each node of the search tree, there has to be a basis for selecting or rejecting a partial solution. As there is no exact way of determining the usefulness of these partial solutions before the end of the algorithm, an upper bound on these solutions has to be evaluated. In most of the branch and bound implementations, linear programming relaxation of 0-1 KP is used to determine the upper bound at each node. With integrality constraint on x_i

and v_i the upper bound derived from Equation 7 is given by Equation 11:

$$U_1 = \lfloor V(CKP) \rfloor = \sum_{i=1}^{c-1} v_i + \left\lfloor \overline{W} \frac{v_c}{w_c} \right\rfloor \quad (11)$$

One of the earliest approach for the exact solution to KP using branch and bound technique was presented by Kolesar [45]. In this algorithm, at each node the item i is selected in the order given by Equation 6. Two branches are formed at each node by fixing x_i equal to 1 and 0. The feasible branch with maximum U_1 value is selected and the search continues. There are many approaches based on some variations of Kolesar's algorithm that are found to be much more efficient. For example, Horowitz and Sahni's [46] algorithm is based on depth first search. Here, the node variable is selected in the same way as Kolesar's algorithm but a greedy strategy is adopted for branch selection. That is, a feasible branch with $x_i = 1$ is selected and the search continues. Martello-Toth [47] algorithm is another effective algorithm based on Horowitz-Sahni strategy. This algorithm uses an improved bound U_2 instead of U_1 and a different dominance criterion to avoid nodes that do not advance the solution.

Greenberg and Hegerich [48] algorithm provides a different strategy for selecting branching variable at each node. Here, the linear relaxation of the induced subproblem is solved and the critical item \tilde{c} is selected as the branching variable. Two branches are created with $x_{\tilde{c}} = 0$, and $x_{\tilde{c}} = 1$. The search continues from the node with $x_{\tilde{c}} = 0$. When the induced CKP has integer solution, the search continues from $x_{\tilde{c}} = 1$. A Matlab function, *bintprog*, is used to demonstrate the application of branch and bound method for the example problem in Table 5. It is a linear programming based branch and bound implementation to solve binary integer programming problems [49]. The basic framework roughly follows the Greenberg-Hegerich algorithm. The algorithm searches for a feasible solution, updates the best solution as the search progresses, and finally verifies that no better integer solution is possible by solving a series of linearly relaxed knapsack subproblems.

As this method can only consider a single objective at a time, Equation 8 is used

to merge three *value* numbers into a single objective to be maximized. Moreover, as the function *bintprog* is designed for minimization problems, the *value* numbers of items are prefixed by a negative sign. The exact solution to the problem with weight constraint W_1 is:

$$x = [1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1]$$

$$V = 53 + 55 + 53 = 161$$

$$\hat{W}_1 = 39$$

Thus, branch and bound algorithm provides a better solution ($V = 161$) as compared to the greedy algorithm ($V = 156$). The solution weight \hat{W}_1 for the former technique is also closer to the constraint than that using the approximate technique.³ The solutions with exact and approximate techniques are graphically compared in Figure 14. The only difference in the greedy and exact solution is in item 10 and 16 where their state is reversed. This follows the observations made by Balas and Zemel [44] that the exact solution of 0-1 KP is very close to that of its CKP counterpart.

Branch and bound technique can also be used for problems with multiple constraints and this is demonstrated by applying *bintprog* on the current problem with constraints W_1 and W_2 . The exact solution is as follows:

$$x = [1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1]$$

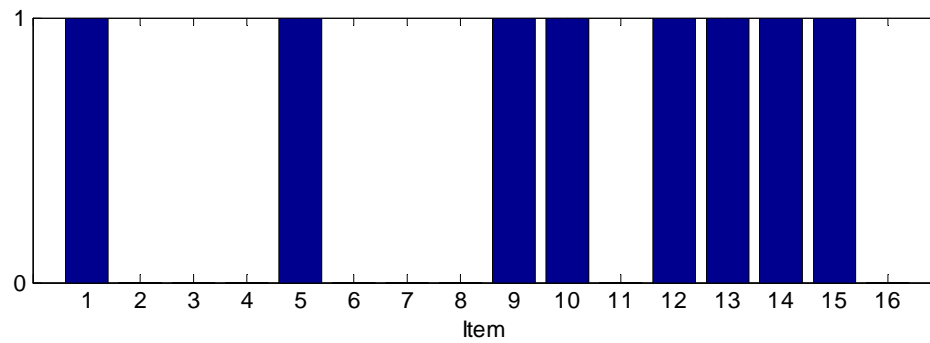
$$V = 49 + 46 + 47 = 142$$

$$\hat{W}_1 = 40$$

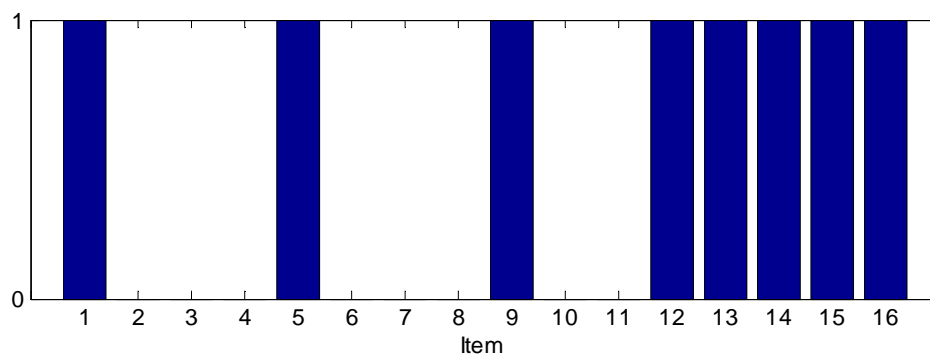
$$\hat{W}_2 = 28$$

There is a corresponding reduction in the value of the knapsack because of the additional constraint. It is interesting to note that constraint W_1 is now active but the value of the knapsack is less than the previous results.

³Though this closeness to the constraint does not necessarily mean that the solution is better.



(a) Greedy Solution



(b) Branch and Bound Solution

Figure 14: Approximate and Exact Solutions to the example KP

3.3.1.1 Advantages and Shortcomings

The main advantage of branch and bound technique is that it provides exact solution to the 0-1 KP. It is a well developed technique with various formulations available to solve large variety of knapsack and similar problems; for example, Elfes, Wesbin and others [24] have used the Martello-Toth [38] algorithm for optimizing technology portfolios. Moreover, these algorithms can also handle multiple constraints.

The branch and bound algorithm can optimize only one objective at a time and this is its main disadvantage for the technology selection problems. When the evaluation of technology combinations is expensive, this technique becomes inviable. It has to traverse considerable number of nodes when there is a large number of technology options available. Moreover, technology selection problems have other inter-technology constraints that can considerably complicate the problem structure on which this technique is based.

3.3.2 Dynamic Programming Algorithms

Dynamic programming (DP) is another enumerative technique typically applied for solving discrete optimization problems and can be used to obtain exact solution to the 0-1 KP. It is a recursive method that combines the solutions to the subproblems to solve the bigger problem. As in *divide-and-conquer* technique [40], this algorithm also divides the problem into subproblems, solve each subproblem optimally, and then combine their solutions to solve the original problem. The only difference between the two being that DP can be applied to the problems whose subproblems are not independent and they share common subsubproblems. The divide-and-conquer technique would repeatedly solve the common subsubproblems and hence work more than required. In contrast, DP algorithm would solve every subsubproblem only once and store the answer, to be used again when required by another subproblem. The 0-1 KP is composed to two main characteristics: optimal substructure, and overlapping

subproblems, that make it amenable to DP implementation.

Optimal substructure property is the Bellman's Principle of Optimality [50] that he stated as:

“An optimal policy has the property that whatever the initial state and initial decisions are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.”

This means that the optimal solution to the problem consist of optimal solutions to the subproblems. In case of 0-1 KP, consider X to be the most valuable knapsack composition with value V and maximum weight W . If we remove item i from this knapsack, $V - v_i$ should be the most valuable knapsack composition weighing $W - w_i$ from the $n - 1$ original items without item i .

The other property that the 0-1 KP has that makes it attractive for DP implementation is of overlapping subproblems. A problem is said to have overlapping subproblems when the recursive algorithm, such as DP, revisits the same subproblem over and over again. A DP algorithm is efficient as it solves each overlapping subproblem only once, storing the solution in a table where it can be looked up when DP revisits the same subproblem. For the 0-1 KP, DP algorithm has to solve the subproblems $V(i, w)$ where, $0 \leq i \leq n$ and $0 \leq w \leq W$. The algorithm consists of populating a two dimensional table with $n + 1$ rows and $W + 1$ columns using the following recursive equation:

$$V(i, w) = \begin{cases} V(i - 1, w) & \text{if } w < w_i \\ \max\{V(i - 1, w - w_i) + v_i, V(i - 1, w)\} & \text{if } w \geq w_i \end{cases}$$

The first row and column are used for initialization and filled with zeros. A pseudocode of a simple DP algorithm for solving 0-1 KP is illustrated in Algorithm 3. A binary array *hold* is used to keep track of items included in the subproblems. This

variable is used to construct the final solution. Kellerer et. al. [51] provide a detailed explanation of this technique for constructing the optimal solution.

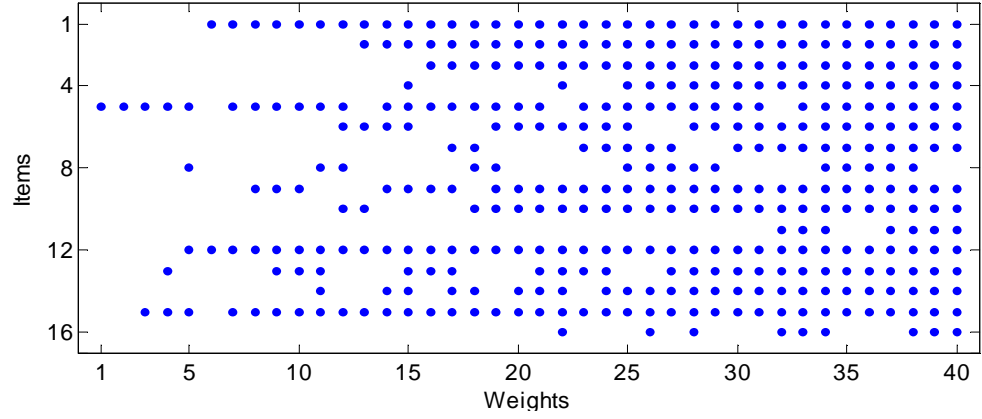
Algorithm 3 Dynamic Programming Algorithm

```

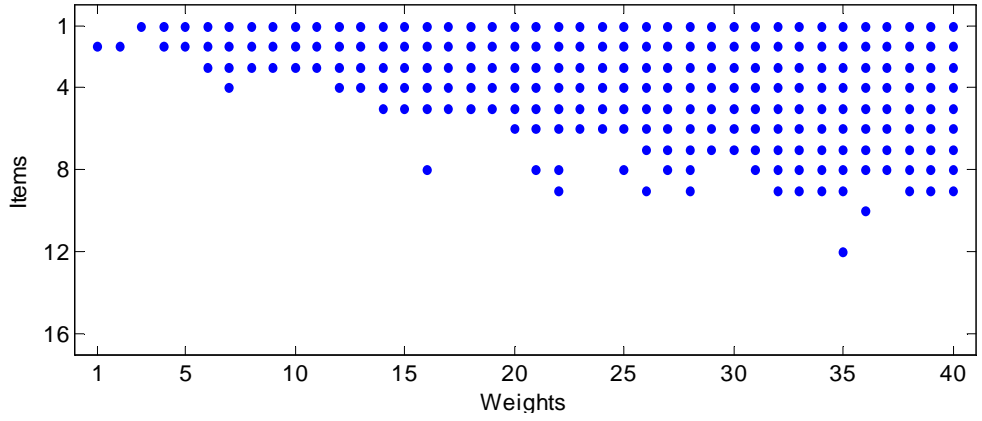
1: procedure DPKP( $n, v, w, W$ )
2:    $V([1 : n + 1], [1 : W + 1]) \Leftarrow 0$ 
3:    $hold([1 : n], [1 : W]) \Leftarrow 0$ 
4:   for  $i \Leftarrow 2, n + 1$  do
5:     for  $w \Leftarrow 2, W + 1$  do
6:       if  $w_{i-1} \leq w$  and  $\{V(i - 1, w - w_{i-1}) + v_{i-1}\} > V(i - 1, w)$  then
7:          $V(i, w) \Leftarrow V(i - 1, w - w_{i-1}) + v_{i-1}$ 
8:          $hold(i - 1, w - 1) \Leftarrow 1$ 
9:       else
10:         $V(i, w) \Leftarrow V(i - 1, w)$ 
11:      end if
12:    end for
13:  end for
14:   $temp \Leftarrow W$ 
15:  for  $i \Leftarrow n, 1$  do
16:    if  $hold(i, temp) = 1$  then
17:       $x_i \Leftarrow 0$ 
18:       $temp \Leftarrow temp - w_i$ 
19:    end if
20:  end for
21:  return  $x, V(n + 1, W + 1)$ 
22: end procedure

```

This DP procedure is applied on the KP of Table 5. As in the previous sections, *values* are merged and W_1 is the constraint on the weight. Being an exact algorithm, the solution obtained is same as from the branch and bound algorithm of the previous section. One noteworthy feature of DP is that it does not require any specific sorting of the items. However, its efficiency improves considerably if the items are ordered according to the Equation 6. This property is illustrated in Figure 15, which shows the sparsity of the *hold* array of Algorithm 3. This variable keeps a record of every new item added to the subproblems, in other words, it keeps track of decisions made to arrive at the optimal solution. Figure 15(a) shows that 392 new items are added in the subproblems while arriving at the optimal solution when the items are not ordered.



(a) With Unsorted Items



(b) With Sorted Items

Figure 15: DP Efficiency With Ordering of Items

However, if the items are ordered, only 233 new items are added while arriving at the optimal solution, as shown in Figure 15(b). Thus, the instances visited by the algorithm are reduced by almost half when the items are ordered according to the value by weight ratio.

3.3.2.1 Advantages and Shortcomings

As in branch and bound algorithm, DP also provides exact solutions to knapsack and similar technology selection problems. But the main advantage of using DP is

that in the process of solving single-capacity KP it also solves all-capacity KP. That is, theoretically one can solve technology selection problem with cost constraint c for all the cost numbers from 0 through c . This is useful when cost is not fixed for a technology selection problem and the decision makers are interested in examining different solutions with changing constraint.

DP is not as efficient as branch and bound method for many problem instances mainly because of its large memory requirements. The run time and memory requirements are dependent on the size of the constraint (in our case W) which can be significantly large when considering, for example, cost constraint for technology selection problems. As a tradeoff, an approximate DP algorithm can be devised by truncating last x decimal digits of the constraint values and thus reducing the total number of subproblems considered by accepting some uncertainty in the result. Other shortcoming of DP, as with many other techniques, is that it can only solve problems with single objective and single constraint dimension.

3.4 Investigative Techniques

There are situations encountered during the development of a system when actual technology selection is not required. Instead, the designers are more interested in investigating the overall combinatorial design space made available by various technology options. These type of situations would normally occur during the early conceptual design phase when the system itself is not fixed and the technologies are evaluated with respect to a generic baseline design.

Two techniques that are applicable for investigating the technology combinatorial space are demonstrated in the following subsections.

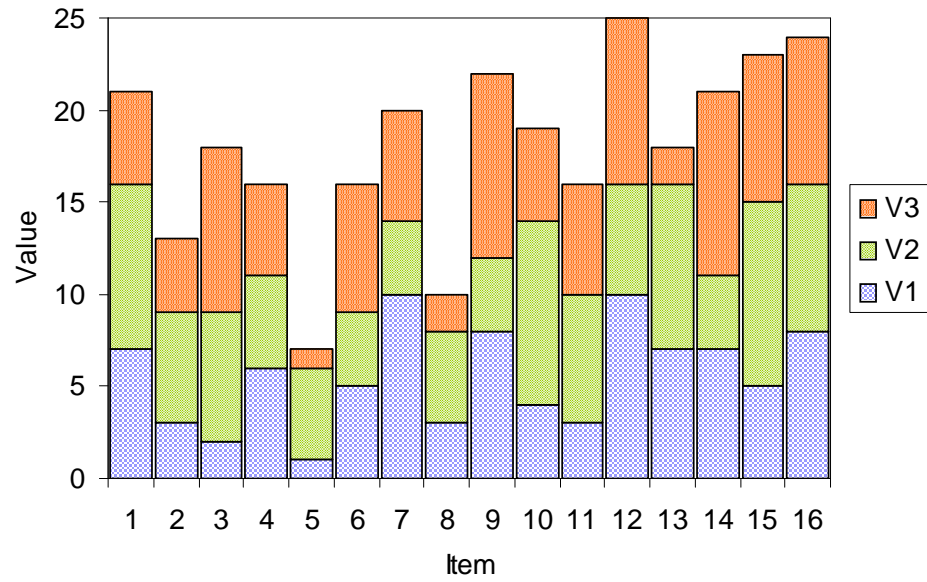
3.4.1 One-On One-Off

One-On One-Off is a preliminary technique that gives a basic idea of the technology options available. It involves examining technology bar charts that are the results

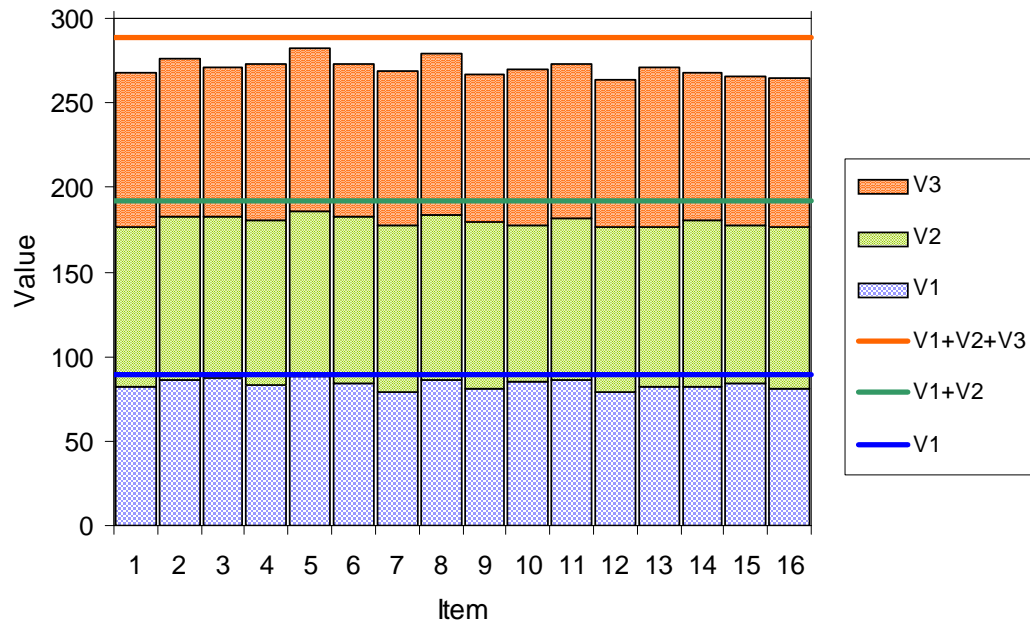
of One-On and One-Off evaluations. One-On evaluation allows designers to compare the impact of each technology on the system level objectives. Here technologies are evaluated with respect to a no-technology-in baseline, that is when no new technology is added to the system. On the other hand, One-Off evaluation allows designers to determine the effect on system metrics of removing individual technologies from the system. In this case the technologies are evaluated from all-technologies-in baseline. That is, for the baseline case all technologies are included in the system, then each technology is removed one at a time and the impact of remaining technologies evaluated. This helps designers understand the importance of a particular technology.

This technique is applied for the KP problem of Table 5 and the results illustrated in Figure 16. The bar charts have item numbers on horizontal axis and values or objectives on vertical axis.

A chart from One-On evaluation is shown in Figure 16(a) and is the most straightforward to interpret. Here, the item represented by the tallest bar has the most impact on the knapsack values. Each section within a bar represents the value of one objective; for this KP we have three objectives, V_1 , V_2 , and V_3 , hence the three sections in each bar. It can be seen that item 12 has the most overall impact followed by 16 and 15; items 5 and 8 have the least impact. The results from One-Off evaluation are illustrated in Figure 16(b). Here, each bar represents the total knapsack value when that particular item is absent and all others are included in the knapsack. Three horizontal lines illustrate the maximum value possible with the bottom one showing maximum V_1 followed by $V_1 + V_2$ and $V_1 + V_2 + V_3$ at the top. In this plot, the most important items are those that show the most degradation or reduction in the values. It can be seen from the figure that the bottom section of item 5 bar almost touches the V_1 line indicating that there is not much to loose in objective V_1 if item 5 is not included. The bars for items 2, 5, and 8 are nearest to the top horizontal line indicating that they have the least overall impact on the knapsack value. While, bars for



(a) One-On Solution



(b) One-Off Solution

Figure 16: One-On One-Off Technique

items 12 and 16 show the most reduction in overall value indicating the importance of these items. From these charts it can be concluded that items 2, 5, and 8 are the least important while items 12, 15, and 16 are the most important. This is verified from the exact solution of Figure 14(b) which has items 12, 15, and 16 present while items 2 and 8 are absent. Though item 5 has minimum overall value it is included in the exact solution; this can be attributed to the presence of constraints that are not considered in One-On One-Off analysis.

These charts can be modified to show a percentage change with respect to the baseline values. The data can also be sorted to show the most effective items or technologies and plotted to view in a Pareto plot.

3.4.1.1 Advantages and Shortcomings

As demonstrated above, One-On One-Off is the most straightforward technique to implement and interpret. It requires minimum time to implement and should be used as the starting step for any technology intensive system design project. This technique can help to identify the best and worst technology options available. The conclusions of this technique can also be used as a sanity check for the final optimized solutions.

Though a single constraint can be included in this analysis by considering the ratio, for example *value/weight* in case of KP, considering multiple constraints is not straightforward. This technique cannot account for the complex constraints involved in a technology selection problem. Moreover, this being a basic technique does not provide any significant details of the technology combinatorial space.

3.4.2 Design of Experiments

Design of experiments (DoE) is a systematic way of conducting formal experimentation. This is widely employed in the fields of biology and social sciences and more recently being used in engineering and economics. The purpose of DoE is to eliminate

correlations (confounding) that exist among the variables and avoid biases. In doing so, it tries to maximize the information gathering potential of each experimental run. This is done by setting up rules and procedures governed by statistics to assign parameter settings on the experimental units. Montgomery [52], among many others, provides one of the most comprehensive discussion on the topic. The experimental designs generated by DoE are characterized by the number of levels used for each parameter. For example, the input parameters in case of technology selection or knapsack problem are characterized by their presence (1) or absence (0), thus two levels for each parameter. In case of *full factorial* design for these problems, there would be 2^n experiments, n representing the number of parameters (technologies or items) under consideration and 2 is for two levels for each parameter. Thus, for the example problem from Table 5, the number of full factorial experiments required would be $2^{16} = 65,536$. This number increases exponentially with increasing items or technologies. Conducting these many experiments may not be feasible nor required for many instances. In such cases, *fractional factorial* designs can be considered. These are also known as screening designs that provide resolution of the main effects of the parameters that are not confounded among themselves or with two-factor (or two-parameter) interactions. They can also estimate two-factor interaction effects that may or may not be confounded with other two-factor interactions. These designs have to be custom created for a given problem and statistical packages such as JMP[®] [53] simplify the task considerably.

To scan the combinatorial design space of the KP under consideration, a fractional factorial design with 64 experiments is implemented in JMP[®]. The resultant 64 item combinations are evaluated for the three *value* and two *weight* responses. A parametric model is generated using least square fitting of the resultant data with items as parameters and total five responses of value and weight. The main output of this

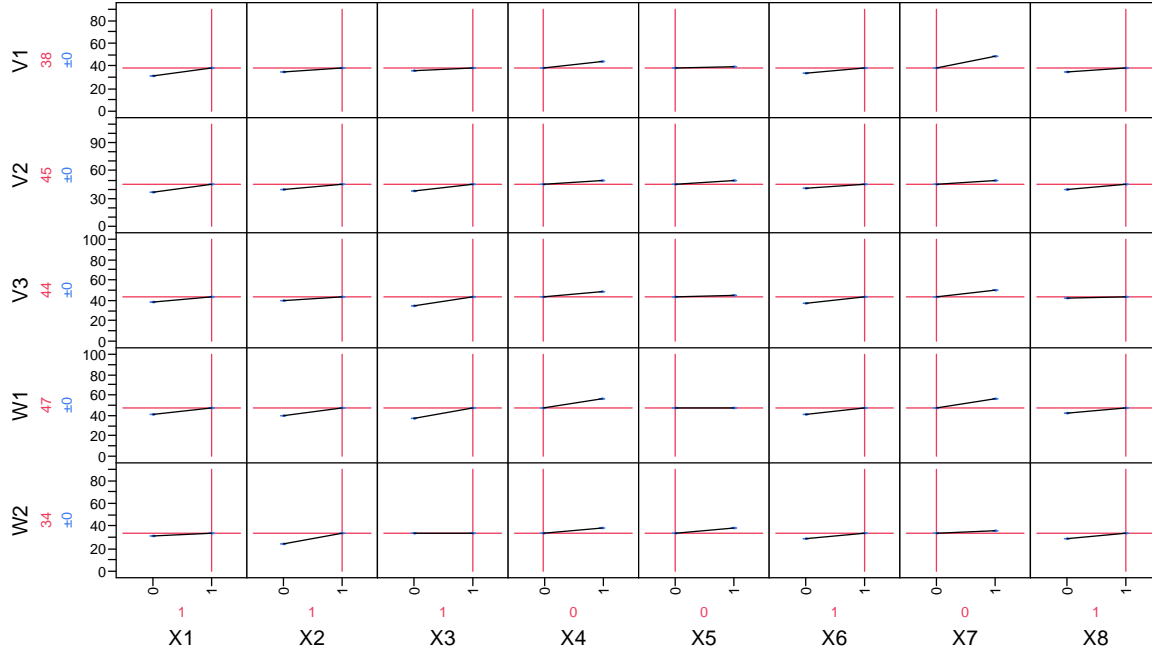


Figure 17: Prediction Profiler for the Knapsack Problem

exercise that is useful for investigating the combinatorial space is a tool called *prediction profiler*. A section of this is shown in Figure 17. Here all the items are present on the horizontal axis and the responses on the vertical axis. The user can set items either on (1) or off (0) and check the resulting values attained by various responses. The slope of line in each cell of the prediction profiler indicates the sensitivity of the corresponding response to the presence or absence of that item in the combination. This is an excellent scanning tool where the user can interactively select a technology and check its impacts on the system. The example of KP is relatively simple than the technology problem. Hence, the response values always increase with the presence of an item. But, more complex interactions present in a technology problem can be investigated by this approach.

A multivariate analysis of variance (MANOVA) can also be implemented on the data generated from the DoE. Similar in concept to single variate analysis of variance (ANOVA) where the samples are divided into groups based on the factors and it is

of the interest to study effect of interactions of these factors on the response. In case of technology problem, the factors are the technologies and they can have two levels, on or off. It is of interest to understand what impacts do technologies have on the responses by themselves or in combination with others. A comprehensive discussion on multi-variate analysis methods is provided by Krzanowski [54] and Manly [55] among others.

3.4.2.1 Advantages and Shortcomings

The DoE based techniques provide excellent tools to investigate overall combinatorial space. Tools like prediction profiler can be created very efficiently by evaluating only a few combinations from many available. It can be very useful to the designer as they can get an approximate idea about the performance of each technology individually and in combination with others. A desirability function is available in JMP® that can perform approximate optimization and provide a good item or technology combination.

One of the main limitations of these techniques is that they cannot be used for selecting a particular combination. As demonstrated with the knapsack example, constraints can be accounted when they are considered as a response. They cannot be defined as constraints as done in other optimization algorithms. Moreover, it is difficult to define technology compatibility and enabling relationships within the framework of statical DoE.

3.5 Advisor for Technology Selection Techniques

Technology problems come in a variety of types. For some problems, there are only a few technology options available and the best combination is to be selected based on its impact on the responses. These responses can be one or more than one. If there are multiple objectives, they can be combined as a weighted sum to form a single objective. There are problems where the aim is not to select the best combination

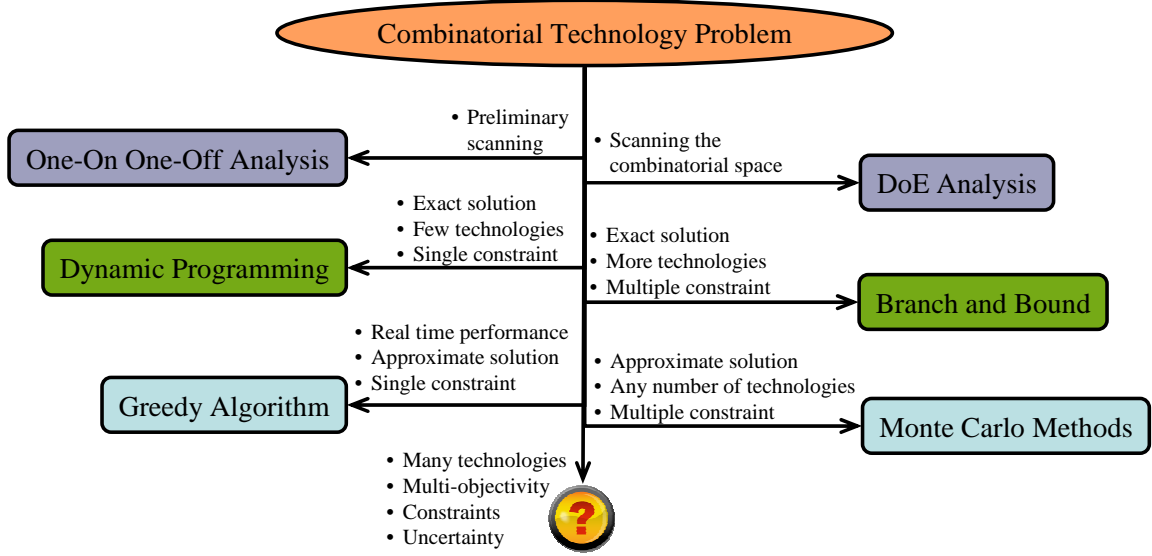


Figure 18: Decision Chart for Technology Selection Methods

but investigate the overall combinatorial space and check what the technologies can do individually and in combination with others. Then there are problems, where there is a large number of available technology options and they impact multiple system responses. The entire combinatorial space has to be investigated for such problems and a best solution selected based on all the responses. Thus, this spectrum is populated with problems having few technologies impacting a single response to many technologies impacting multiple system responses in a complex manner. To address this wide variety of technology problems and select a suitable technique or algorithm for technology selection, an *advisor* is created. It is based on the algorithms and techniques discussed above and is in the form of a decision chart as shown in Figure 18.

Once a combinatorial technology problem is defined, a technique or an algorithm is selected to solve it based on the problem characteristics and the main purpose. For setting up the problem for quantitative analysis, it is suggested to use the technique based mapping technology impacts to technology metrics of the system model. This framework is used in the Technology Identification, Evaluation and Selection

(TIES) method described in the previous chapter. For qualitative analysis, the link between customer requirements and technology options can be modeled using inter-linked decision or planning matrices as used in the Strategic Prioritization Process (SP2) described in the previous chapter.

If the main purpose of the qualitative or qualitative study is of preliminary scanning of the technology options available, a one-on one-off approach can be implemented. It is a simple analysis conducted to study the system level impacts of the presence and absence of individual technologies. The technologies are analyzed in isolation and compatible and enabling relations among them are ignored. If the scanning to the combinatorial technology space is desired, DoE analysis can be implemented. A DoE is created using one of the screening designs and the technology combinations are evaluated qualitatively or quantitatively. A prediction profiler can be created as explained previously and the system level technology behavior can be examined. Any technology combination can be analyzed in real time with this technique.

If the main aim of technology problem is of optimization and an exact solution is required, an exact algorithm has to be implemented. When the number of technologies under consideration is not too high,⁴ and only one constraint is considered, dynamic programming can be used. Multiple objectives can be considered using a weighted sum approach.⁵ If on the other hand, there are multiple constraints in the problem, a branch and bound approach can be implemented. This can handle multiple constraints and is faster than the dynamic programming. Multiple objectives can be considered using a weighted sum approach as in dynamic programming. For a problem that requires an exact solution, dynamic programming should only be considered if one is interested in the solutions to the subproblems; that is, if the main

⁴In the range of 20-30 depending on the time it takes to evaluate a technology combination.

⁵More on this in Chapter 4.

problem has constraint $cost = C$ and one is also interested in solutions to the sub-problems with $cost = 0$ through C , dynamic programming should be used. For all other instances of exact optimization, branch and bound is a better choice.

If, on the other hand, the goal of the problem is technology optimization and the number of technologies available is very large, arriving at the exact solution may not be possible with available computational resources and time. Moreover, in many instances of technology problems, exact solutions are seldom required. In such situations approximate algorithms should be considered. If realtime performance is required and the problem has single constraint, a greedy algorithm is the best option. This is considerably fast for a large number of technologies. Multiple objectives are considered using the weighted sum approach. Weights for the objectives can be changed, new ranking assigned to the technologies based on these weights, and the best combination selected based on the ranking, all within seconds on a desktop computer for problems with 200-300 technologies. If the realtime performance is not required, Monte Carlo method can be used for solving the problem approximately. The benefit of using this techniques is that multiple system level constraints can be considered. This is also the preferred approximate technique if compatibility and enabling constraints among the technologies have to be considered.

The techniques and algorithms described above have a common limitation. They can handle only one objective. When the technology problem has multiple objectives, they all have to be lumped into a single objective using a weighted sum or related technique. Not all of them handle multiple system level constraints or inter-technology constraints (compatibility and enabling constraints). Moreover, these algorithms cannot consider uncertainty associated with the technology impacts. Thus a method to address these limitations is required.

3.6 *Summary*

Various algorithms and techniques that can be used for technology selection problems have been discussed in this chapter. To demonstrate these techniques, a benchmark problem was defined. This problem is a multi-objective multi-constrained knapsack problem that is shown to be a simplification of the combinatorial technology selection problem. Three classes of techniques have been explained: investigative technology scanning techniques, exact algorithms, and approximate algorithms. Two examples for each class are demonstrated. Based on these techniques, an advisor for choosing an appropriate method for a technology problem is presented.

It is noted that the algorithms and techniques described in this chapter can handle only one objective. If multiple objectives are present, they have to be combined into a single one using a weighted sum or similar technique which have significant limitations as will be described in the next chapter. These techniques cannot account for technological uncertainties. Moreover, only a few can consider multiple system constraints or inter-technology constraints. Thus a comprehensive method is required that can address technology problems with large number of technology options, multiple objectives, multiple constraints, inter-technology constraints, and technological uncertainties. The following chapters of this thesis will describe the quest for such a method.

CHAPTER IV

MULTI-OBJECTIVE DECISION MAKING

One of the main aspect of the problem at hand is about Multi-Objective Decision Making (MODM). Given t number of technologies and n objectives, one has to decide on the best combination of technologies that satisfies all the requirements. MODM, in contrast to Multi-Attribute Decision Making (MADM), is associated with design problems; here, it is required to *design* and *select* the *best* alternative that satisfies all the constraints and meets all the requirements. Looking from this perspective, Multi-Objective Optimization (MOO) is an intrinsic part of MODM techniques. MADM on the other hand deals with only *selection* of the *best* alternative from an existing set of options described by their attributes. Thus if there are only a handful of technology combinations that are being considered, a MADM approach can be adopted for selecting the best alternative. On the other hand, if the scope of the problem is too big, MODM techniques have to be explored.

This chapter investigates some of the classical and more recent MODM approaches. Limitations with the classical techniques are explained and a family of non-domination based techniques is described that can help eliminate these shortcomings. One of the main challenge with this technique is regarding redundant dimensions in the problem formulation. Two techniques to address this are compared using a benchmark knapsack problem. Other challenges with the non-domination based technique are also discussed.

4.1 MODM Approaches

The final solution of a multi-objective problem is the result of both *decision* and *optimization* processes [56]. Decisions in such problems are anchored around the

preferences of decision makers (DMs). The compromises they make among various objectives, in addition to the problem constraints, define a region of interest in the multi dimensional solution space. The DMs express their preferences towards various objectives, to an analyst or a computer program, at some specific point during the MODM process. Hwang and Masud [57] classify various MODM methods based on the preference information from decision maker (DM) known before, during or after the optimization process. These are stated below:

1. *No Articulation of Preference Information*: Here DM is not required to define any particular preference information after the problem is set up with constraints and objectives. But in doing this, the analyst or the optimization program may have to make some assumptions about DM's preferences. Moreover, DM should be able to accept the solution offered by this process.
2. *A Priori Articulation*: Preference information is given by the DM to the analyst before solving the problem. This information can be in the form of a weighting or preference vector for the objectives. If correctly used, this can ensure the most satisfactory solution to the DM. One of the main drawback of this technique is that the preferences are articulated in information vacuum.
3. *Progressive Articulation*: This is the class of interactive methods. Here, the DMs decide on their preferences based on the current solutions as the search progresses. There is a feed back loop between the DMs and analyst/machine. With these techniques DM is part of the solution and in the process learning about the problem. Much more effort and time are required on the DMs part as they are intimately involved in the process.
4. *A Posteriori Articulation*: In this class of methods, MODM is divided into two distinct phases. In the first phase, a subset of non-dominated solutions in the objective space is determined. Next, the DMs make implicit tradeoffs between

objectives based on some criteria, which may be non-quantifiable, and choose the most satisfactory solutions from the given subset. This technique does not require the DMs to express their preferences beforehand in information vacuum, on the other hand, it does generate a large number of non-dominated solutions.

The past implementation of optimization routines for technology selection as seen in Chapter 2 and 3 belong to the *A Priori preference articulation* class of MODM methods. The following discussion is to investigate this technique in detail and illustrates its limitations.

4.1.1 *A Priori* Preference Articulation

As stated before, technology selection is a multi-objective optimization problem. There are various objectives, often conflicting, to optimize and some form of compromise is essential. A simplified mathematical representation of this problem is defined by Equation 12.

$$\text{minimize} \quad F(x) = (f_1(x), f_2(x), \dots, f_n(x))^T \quad (12)$$

$$\text{subject to} \quad g_i \leq 0, \quad i = \{1, 2, \dots, m\} \quad (13)$$

Here, x is a binary vector of length t that states the presence (1) or absence (0) of technologies. There are n objectives in this problem and the most straightforward approach is to convert them into a single objective. This approach is known with a variety of different names such as *Utility Function*, *Aggregation*, *Scalerizing*, *Weighting* or *Overall Evaluation Criteria (OEC)* based method. For this, a weight vector w of length n is considered with $0 \leq w_i \leq 1$ and $\sum_{i=1}^n w_i = 1$. The values of w_i are fixed by DMs before the optimization process. This formulation transforms the problem from minimization of n functions of Equation 12 to that of a single function as shown in Equation 14.

$$\text{minimize} \quad y = w^T F(x) = \sum_{i=1}^n w_i f_i(x) \quad (14)$$

For simplicity, the following discussion is based on a two dimensional problem; it can be readily extended for multiple dimensions. Rewriting Equation 14 for two dimensions:

$$y = w_1 f_1(x) + w_2 f_2(x)$$

and after rearranging, we have:

$$f_2(x) = -\frac{w_1}{w_2} f_1(x) + \frac{1}{w_2} y \quad (15)$$

This is an equation for a line where $-\frac{w_1}{w_2}$ is the slope and $\frac{y}{w_2}$ is the intercept on the vertical axis.

Das and Dennis [58] provide an excellent trigonometric treatment of the two dimensional multi-objective problem. From this geometrical perspective, the minimum value for Equation 14 is determined by moving the line given by Equation 15 in a perpendicular direction to itself in the objective space. This is illustrated in Figure 19. The optimal point is where this line is tangential to a curve. This curve is known as the *Trade-Off Curve* or the *Pareto Frontier* and is the locus of all the Pareto-optimal points in the objective space. Thus the optimal point obtained by this method is also a Pareto-optimal point. The Pareto-optimal solutions are also known as non-dominated solutions, efficient solutions, or non-inferior solutions in the literature [57, 59].

Definition: A point \bar{x} is said to be **Pareto-optimal** if and only if there does not exist another point x in the design space such that for every $i \in \{1, \dots, n\}$, $f_i(x) \leq f_i(\bar{x})$ and for at least one $i \in \{1, \dots, n\}$, $f_i(x) < f_i(\bar{x})$.

The Pareto frontier exists in the objective space because no single \bar{x} can minimize all the objectives at the same time. When the Pareto front is convex, the entire curve can be generated using weighting method. For a two dimensional problem, the weights for two objectives can be represented by a single quantity α . Let $w_1 = \alpha$; this

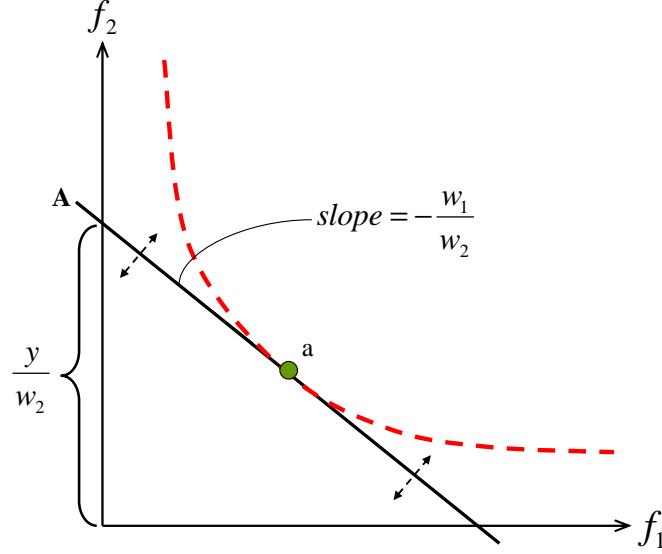


Figure 19: Geometric Interpretation of Weighting Method

implies $w_2 = 1 - \alpha$. Now, by changing α from zero to one, one can generate a series of lines with varying slopes that result in different non-dominated points on the Pareto front as illustrated in Figure 20. Here, each line **A**, **B**, **C**, and **D** is generated using different values of α (α_a , α_b , α_c , and α_d respectively) and they provide corresponding non-dominated points in the objective space.

4.1.1.1 Limitations of A Priori Preference Articulation

The previous discussion showed that one can obtain all the points lying on a *convex* Pareto frontier using the weighted sum based methods. But what if this front is non-convex? This situation is shown in Figure 21. In this figure, line **A** associated with a certain α_a value is tangent to two points, a and c , on the Pareto front. Hence, there are two optimal points corresponding to that particular α_a value. This is an indication that the Pareto front has a non-convex section. Now, let's consider that a point b , lying within the non-convex section ac of the curve is of interest to the DM. The tangent to this point b is a line **B** with slope defined by α_b . As seen in the figure, the line **B** intersects the Pareto front at point p and is not tangent to the curve here.

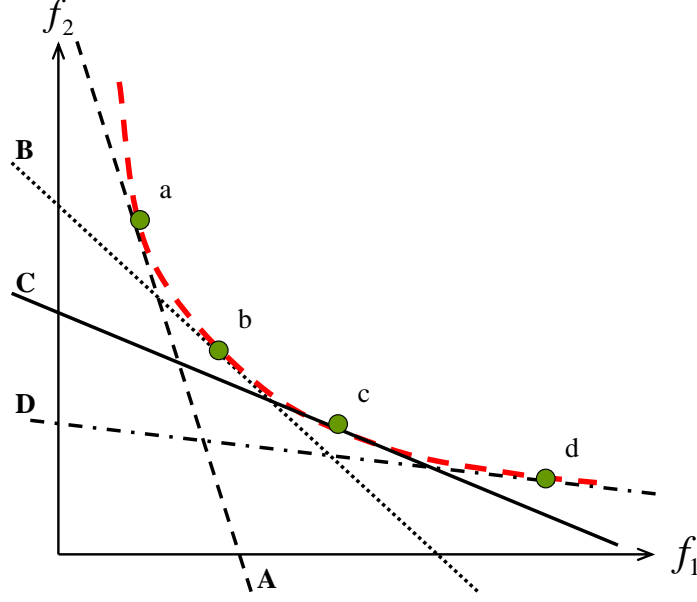


Figure 20: Generating Pareto Frontier Using Weighting Method

Thus, for the given weighting, the optimal point can be better than b . This can be achieved by moving the line **B** down perpendicularly to itself. This is indicated by line **D** in Figure 21 and d is the optimal point for the weights corresponding to α_b . This is the case with any point within the section ac ; if the slope of line **B** is greater than that of line **A** (considering the negative slope), the optimum lies beyond point c and if slope of **B** is less than that of **A**, the optimum lies beyond point a . Hence, it is impossible to obtain points within a non-convex section of the Pareto frontier with any combination of weights using the weighting method.

Apart from not being able to find points in the non-convex section, the weighted sum method is also unable to find evenly spaced points on the Pareto front given evenly spaced weights. This is another significant drawback that is highlighted by Das and Dennis [58] in their critical examination of the weighting method. Thus for a two dimensional problem, an even spread of α may not result in an even spread of non-dominated points on the Pareto front. The consequence of this property is that depending on the shape of the Pareto front, an equal weighting on all objectives may

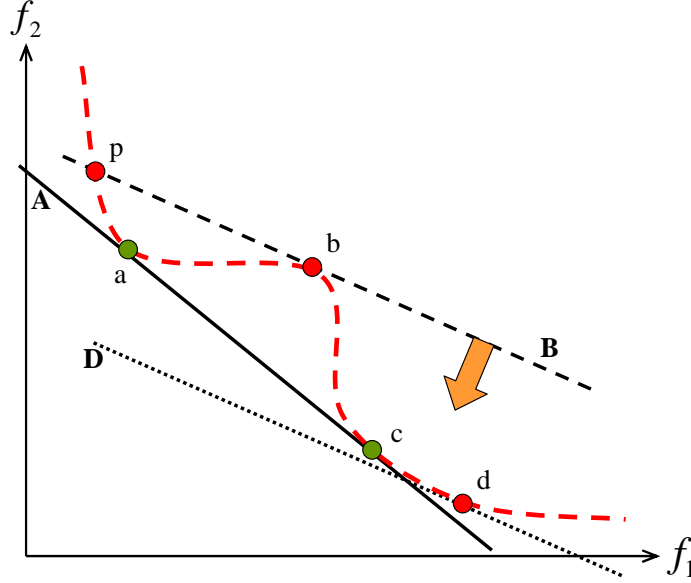


Figure 21: Weighting Method with Non-Convex Pareto Frontier

not result in a point on the center of the front. This is illustrated in Figure 22. Here the DM, by giving equal weighting to f_1 and f_2 , expects a solution in the middle of the Pareto front. But because of the skewed shape of the front the solution lies in its upper region. It is impossible to guess the weights required to obtain results in the region interest of the Pareto front without a prior knowledge of the shape of this front. Hence, even if the Pareto front is convex, the weighting method is not very effective as the DMs have to make tradeoffs among objectives and fix weights without prior knowledge of the shape of the trade-off curve.

Discussion in the above paragraphs revolved around shortcomings with mathematical aspect of the weighting method, mainly related to the shape of the Pareto front. But there is another aspect that requires due attention and that forms the basis of various utility function based methods— elicitation of *weights*. Here, weights are elicited from the DMs prior to the optimization process. It is important to ensure independence or orthogonality of the objectives before assigning weights. These

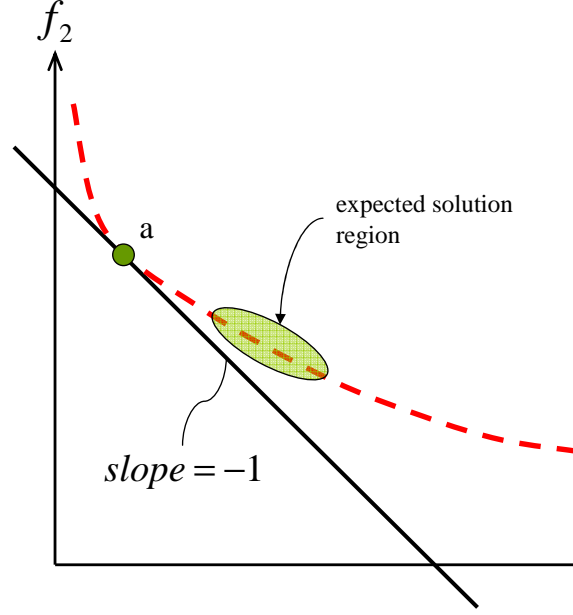


Figure 22: Weighting Method with Skewed Pareto Frontier

weights are based on the relative importance of each objective and are often determined via pair wise comparison of various objectives. This becomes very complicated as the number of objectives increases. Hazelrigg [60] observes that for multi-attribute design problems, weights can only be accurately articulated by comparing the end products and not on the basis of comparing attributes, or for that matter objectives alone. Moreover, there is considerable amount of uncertainty involved, both in technology impacts and the objectives themselves, during the early design phases that it is difficult to pin down exact weights for the objectives. In other words, it would be very difficult for the DMs to accurately assign relative weights to the objectives at an early design phase, more so without looking at available design options in the objective space.

Recapitulating above discussion, the drawbacks with *a priori* preference articulation or utility function based methods are as follows:

- Impossible to capture non-convex part of the Pareto frontier.
- Impossible to predict weights that result in optimized points in the region of

interest without prior knowledge of the front.

- Difficult for DMs to assign weights and make tradeoffs without knowing all the options available.

It is clear from the above discussion that *a priori* preference articulation based methods are not suitable for the multi-objective technology selection problem.

4.1.2 *Progressive* Preference Articulation

The other option is the *progressive* articulation of preferences. Parmee et al. [61] and Buonanno [62] have implemented evolutionary algorithm based interactive optimization methods for conceptual aircraft design. In these methods, the designer is involved in the optimization process and guides the search towards a region of interest on the Pareto front by providing preferences at intermediate stages.

These methods do remedy some of the drawbacks of the weighted sum based methods such as the difficulty of assigning weights. But, there is an associated penalty in form of DMs effort and time. The DMs and designers have to be present in front of the computer when the optimization is carried out and depending on the problem, this process may take a long time. Hence, this too may not be an ideal solution for the technology selection problem.

4.1.3 *A Posteriori* Preference Articulation

This brings us to the *a posteriori* preference articulation — the last class of MODM techniques described by Hwang and Masud [57]. As mentioned before, these methods involve first identifying a subset of points that populate the Pareto front (or hypersurface in multi-dimensional space) and then making tradeoffs between various objectives and selecting the most suitable point. The DMs are only involved in making the tradeoffs and the final selection. The search for non-dominated points is carried out without requiring their presence. Various search techniques used within

this method manage to overcome the drawbacks associated with *a priori* preference articulation. Moreover, as the DMs are only involved in making tradeoffs and final selection, these methods are much more efficient for them when compared with *progressive* preference articulation.

The implementation of *a posteriori* preference articulation is a challenging task, especially when the dimensionality of the problem is large. These challenges include the requirement of significant computational resources, the existence of large number of non-dominated points, and difficulty of making tradeoffs in more than three dimensions. Though these are serious concerns, effective tools and techniques are available to mitigate them. For example, the tremendous increase in computational power of a desktop computer in the last decade and the development of evolutionary algorithms to search for the Pareto frontier has considerably reduced the computational time required for the task [63, 56]. Deb and Saxena [64], and Brockhoff and Zitzler [65] have successfully demonstrated techniques to reduce the dimensionality of the problem in the context of Pareto optimization. Horn [59] and Zitzler [66] among others have explored techniques of niching and clustering respectively to obtain an even distribution of points along the Pareto front. With such techniques, only a fraction of Pareto optimal points can accurately represent the tradeoff surface. Moreover, the availability of commercial visualization and analysis tools such as JMP® [53] has made the task of making implicit tradeoffs in multi-dimensional objective space relatively easy for the DMs.

In the light of these observations and because of the limitations of the *a priori* and *progressive* preference articulation frameworks, the following hypothesis is proposed addressing the question: *How to address the multi-objective nature of the technology selection problem?*

Hypothesis: *A Posteriori* preference articulation, a class of MODM methods, can be used to address multiple objectives in the technology selection problem and

identify a satisfactory solutions.

This can be considered as a high level hypothesis that will pose further questions. The following sections investigate the core of this method and corresponding research questions in detail.

4.2 *Pareto Optimality*

By definition, multi-objective problems do not have a single answer. There is a tradeoff involved among objectives and depending on different preferences, one can have different answers. Thus, the concept of global optimization is not well defined for such problems [56]. In this context, *Pareto Optimization* can be thought of as a meaningful way of global optimization of the multi-objective problem.

Definition: Pareto Optimization is the process of searching for a subset of non-dominated or Pareto optimal solutions in a multi-dimensional objective space.

It is important to note that the Pareto optimized solution set is a *subset* of the set of all non-dominated solutions. We are generally interested in only a subset because the cardinality of the set of all non-dominated solutions can be infinite for a continuous problem. For the combinatorial problem addressed in this thesis, this cardinality, even though finite, is extremely large. Thus, depending on the granularity or resolution or density of the Pareto front, each tradeoff is represented in the Pareto optimized solution set.

Pareto optimization is the first step for *a posteriori* preference articulation methods. For this, it is essential to have a clear understanding of the concept of non-domination and Pareto optimality. According to the definition, within a Pareto optimal solution set, no objective function can be improved without a simultaneous deterioration in at least one of the other objectives. This concept is eloquently explained by Zitzler [67] and can be visualized for a hypothetical two dimensional

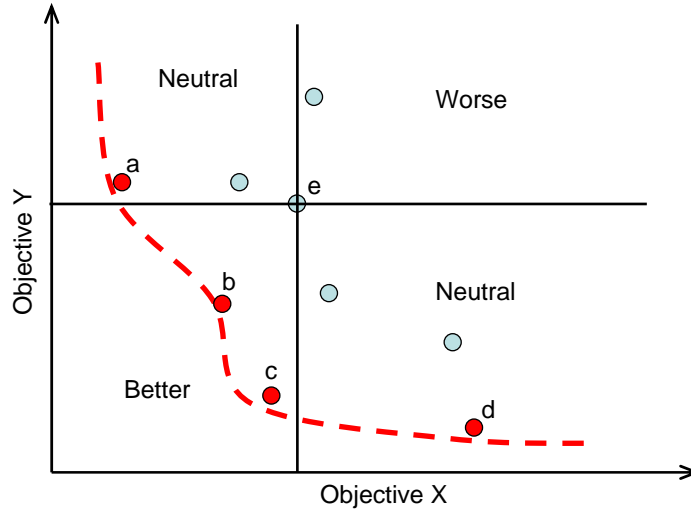


Figure 23: The Concept of Pareto Optimality

minimization problem in Figure 23. The figure shows all possible solutions in a two dimensional objective space. To evaluate the domination condition of point e for example, the objective space is divided into four quadrants with e as the origin. The points lying in the upper-right quadrant are considered as being dominated by e . The points in the lower-left quadrant are said to dominate e ; in other words they are not dominated by e . The points in other two quadrants are neutral with respect to e ; they have no bearing on the domination condition of e . Evaluating all the points in the objective space for their domination condition, points a, b, c and d are found to be non-dominated with respect to all the points in the space. Thus they are the Pareto optimal points and form the Pareto frontier. The same logic can be extended to evaluate the domination condition and find Pareto optimal points in n dimensional space.

4.3 Challenges with *A Posteriori* Preference Articulation

As mentioned earlier, there are difficulties involved in implementing *a posteriori* preference articulation for large dimensional problems. Most of the difficulties arise because of the increasing dimensionality of multi-objective problems. If there are only two or three objectives to optimize, the Pareto front is manageable and its very intuitive for the DMs to make tradeoffs. As the number of objectives increase, the scale of Pareto front increases, the task of finding representative non-dominated points becomes difficult, and making tradeoffs also becomes difficult.

For a two objective problem, the Pareto front is represented by a set of points along a one dimensional curve. In general, for a problem with n objectives, the Pareto front is a hypersurface with $(n-1)$ dimensions. Thus as we move from dual to multiple objectives, the dimensionality of the Pareto front increases, and the points required to represent the front increase considerably. This is verified by a simple experiment where 5000 random points are considered in a 15 dimensional space. Non-dominated points are extracted for two through fifteen dimensions. A ratio of the number of non-dominated points to the total points (5000) is plotted in Figure 24. As seen from this plot, there is a rapid increase in the number of non-dominated points with increase in the dimensions. With a fifteen dimensional objective space, almost all of the 5000 points are Pareto optimal.¹ This explosive increase of non-dominated points in higher dimensions is the root cause of most of the problems faced in this method.

With this observation, three main questions arise that have to be addressed for an effective implementation of the *a posteriori* framework. They are:

- How to reduce the dimensionality of a multi-objective problem in the context of Pareto optimization?

¹These results are with a random sampling, the proportions may differ for an actual technology problem.

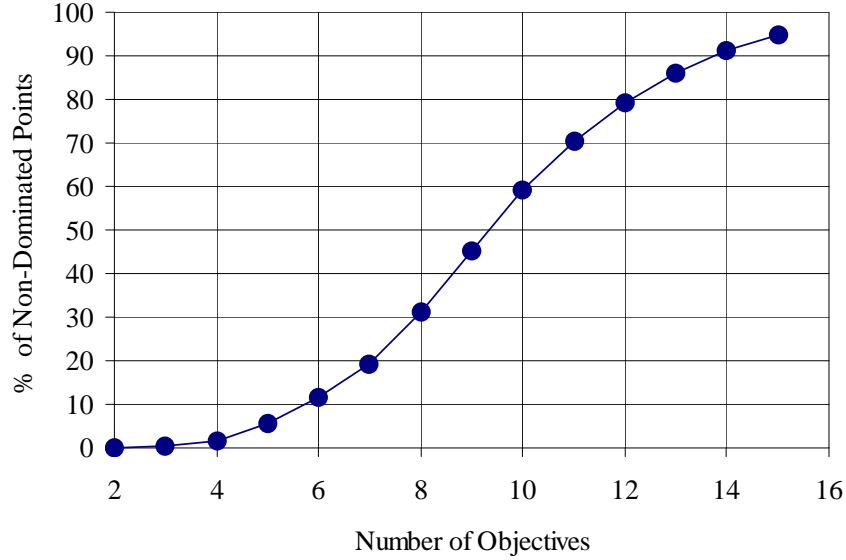


Figure 24: Increase in Proportion of Non-Dominated Points with Dimensions

- How to reduce the total number of points required to represent a multi-dimensional Pareto front?
- How to efficiently search for these non-dominated points in multiple dimensions?

The last two questions are generally intertwined. Almost all the well known methods used for Pareto optimization attempt to generate a subset of Pareto optimal points that has even distribution along the front; in the process reducing the total number of points required to represent the front. The following subsections take a closer look at the challenges posed by the above questions.

4.3.1 Reducing the Dimensionality of Pareto Frontier

One way of looking at this problem is to investigate if the dimensionality of the Pareto front for an n dimensional problem is really $(n - 1)$. As Veldhuizen [56] has shown, the dimension of the Pareto front is *at most* $(n - 1)$ — it can be lower than that. Thus, if the Pareto front dimensionality is lower than $(n - 1)$, a question arises — Are all n objectives necessary? This is a fairly recent direction of research and there are two basic approaches available for addressing the issue.

Table 6: Example Multi-Objective Problem

	Objectives				
Items	v_1	v_2	v_3	v_4	v_5
1	7	-9	5	-6	3
2	3	-6	4	-7	10
3	2	-7	9	-10	1
4	6	-5	5	-9	4
5	1	-5	1	-1	4
6	5	-4	7	-6	5
7	10	-4	6	-10	2
8	3	-5	2	-5	5
9	8	-4	10	-7	3
10	4	-10	5	-6	7
11	3	-7	6	-6	6
12	10	-6	9	-5	2
13	7	-9	2	-4	10
14	7	-4	10	-5	7
15	5	-10	8	-3	6
16	8	-8	8	-8	8

4.3.1.1 Based on Principle Component Analysis

The first approach is by using the Principle Component Analysis (PCA) as proposed by Deb and Saxena [64]. Their method aims at retaining the objectives that can explain most of the variance in the data. PCA is one of the simplest multivariate analysis technique and is explained in most of the textbooks on the subject [55, 54, 68]. PCA based dimensionality reduction for a multi-objective problem is best understood with the help of an example. Lets consider a 16 item knapsack problem as studied in Chapter 3. The problem considered is described in Table 6. Here each item has five assigned objective values and no constraint is considered. The problem is to find a combination of items to:

$$\text{minimize} \quad V_i = \sum_{j=1}^{16} v_{ij}x_j$$

$$\text{where} \quad x_j \in \{0, 1\}, \quad i \in \{1, 2, 3, 4, 5\}, \quad j \in \{1, 2, \dots, 16\}$$

Let the number of observations or item combinations considered are n . For this

Table 7: Correlation Matrix \mathbf{R}

	V_1	V_2	V_3	V_4	V_5
V_1	1.000	-0.842	0.901	-0.885	0.772
V_2	-0.842	1.000	-0.847	0.865	-0.894
V_3	0.901	-0.847	1.000	-0.905	0.750
V_4	-0.885	0.865	-0.905	1.000	-0.784
V_5	0.772	-0.894	0.750	-0.784	1.000

example, all 2^{16} combinations are considered; PCA is carried out on $65,536 \times 5$ matrix. The first step of the process is to standardize the data. This is done by considering each dependent variable separately (each column of the matrix), subtracting the mean of this variate from each observation, and dividing the resultant value by the variate's variance. The standardization helps in making the data comparable in terms of value and variance. With this data, correlation matrix \mathbf{R} is computed; Table 7 lists this matrix for the example problem. It can be observed from this matrix that the set of V_1 , V_3 , and V_5 are positively correlated with each other and also the set of V_2 , and V_4 are positively correlated. On the other hand, V_1 , and V_2 are negatively correlated, they are conflicting. In fact, each of the variable of one set is in conflict with each of the variable of the other set. Thus, any one variable may be selected from each sets to approximately represent the solution space. Now, questions such as — How to choose a variable from each set? Are only two variables enough? — have to be answered. To address these questions and to analyze more complex and higher dimensional technology selection problems, this statistical analysis has to be extended towards a PCA based approach as suggested by Deb and Saxena [64].

For PCA, the eigenvalues for the correlation matrix \mathbf{R} are calculated and ranked in the decreasing order of their magnitude. Figure 25 shows the ranked eigenvalues along with a Pareto plot showing the percent contribution of each value to the sum of all eigenvalues. The eigenvectors corresponding to these eigenvalues are listed in Table 8; each column representing one eigenvector denoted by PC_i . These eigenvectors give the principle components of the new objective space. And, the percentage

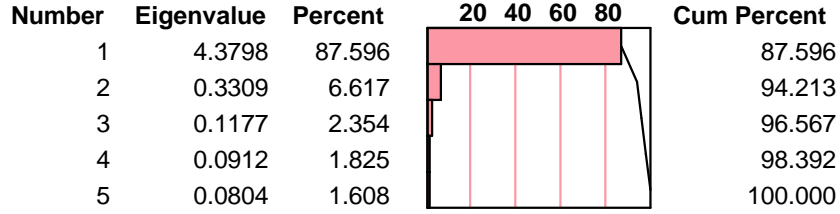


Figure 25: Eigenvalues for **R**

Table 8: Eigenvectors Corresponding the Eigenvalues

PC_1	PC_2	PC_3	PC_4	PC_5
0.4497	0.3370	0.7570	-0.1953	-0.2703
-0.4541	0.3401	0.2373	-0.4424	0.6528
0.4501	0.4213	-0.1192	0.5738	0.5258
-0.4537	-0.2691	0.5680	0.6301	0.0451
0.4280	-0.7217	0.1839	-0.1996	0.4715

of eigenvalue shown in Figure 25 represents the proportion of total variance explained by the corresponding eigenvector. For the current example, PC_1 can explain about 88% of the total variance in the data set. Similarly, the first three principle components can account for about 97% of the total variance. The elements of eigenvectors are the coefficients used to form the linear combination of original variables, creating the principle component variable. Thus, each element of the eigenvector represents the relative contribution of the respective objective or dependent variable. In PC_1 , the contribution of V_1 is 0.4497, contribution of V_2 is -0.4541 , and so on. The objectives that contribute the most to the principle component variable are the ones corresponding to the most positive and the most negative element of the eigenvector. Thus, by analyzing higher ranked principle components in this manner, one can select the most significant objectives.

When there are large number of objectives, Deb and Sexena [64] suggest using a predefined threshold cut (TC). The top ranked principle components with cumulative contribution greater than or equal to TC are selected for analysis. Significant

objectives are then extracted from these selected principle components. For the current example, a higher TC of 95% is chosen and from Figure 25 it can be observed that PC_1 , PC_2 , and PC_3 fall within this threshold. Analyzing the eigenvectors from Table 8, objectives V_2 , and V_3 are selected from PC_1 ; V_3 , and V_5 are selected from PC_2 ; and V_1 , and V_3 are selected from PC_3 . Thus, a five dimensional objective space is represented by the four most important dimensions. To investigate if further reduction is possible, the correlation matrix for only the selected variables is examined. This matrix is same as R listed in Table 7 with the row and column for V_4 removed. As observed from this correlation matrix, V_1 , V_3 , and V_5 are closely and positively correlated. Thus, V_1 can be considered redundant and de-selected as it was the last objective to be selected from the third principle component. This leaves 2nd, 3rd, and 5th objective out of the total five. This is the same result as one would obtain if only first two principle components were selected.

Deb and Saxena [64] have suggested an iterative procedure of using the PCA based analysis in conjunction with a Pareto searching algorithm to reduce the dimensionality. It can be safely assumed that one of the reasons behind the authors suggestion of iterative implementation of the analysis is the possibility that the Pareto front may have different statistical properties than a set of randomly selected points from the objective space. This may lead to difference in identifying the important objectives depending on where the sample points are chosen from — entire objective space or the Pareto front. To verify this, a complete set of non-dominated solutions (4,005 points) is extracted from the entire combinatorial space of 65,536 points. Now, the PCA based analysis is executed using these Pareto solutions as sample points. The eigenvalues and eigenvectors for this analysis are shown in Figure 26 and Table 9 respectively.

Defining TC at 95% as before and conducting the same analysis, V_1 , V_3 , V_4 , and V_5 are selected as significant objectives. Trying to further reduce these dimensions

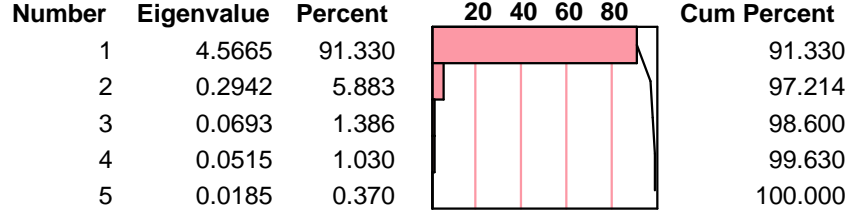


Figure 26: Eigenvalues for Non-dominated Sample Points

Table 9: Eigenvectors for Non-dominated Sample Points

PC_1	PC_2	PC_3	PC_4	PC_5
0.4520	0.3344	0.0020	0.8036	0.1950
-0.4536	0.3110	-0.6127	-0.0104	0.5675
0.4497	0.4108	0.3279	-0.5409	0.4784
-0.4542	-0.2671	0.6498	0.2463	0.4894
0.4258	-0.7425	-0.3080	-0.0302	0.4142

by observing the correlation matrix reveals that either V_1 or V_3 can be selected. V_1 is retained as it was selected first through PC_1 . Thus V_1 , V_4 , and V_5 are selected by considering only the Pareto optimal points. This is in contrast to V_2 , V_3 , and V_5 that were selected by considering all the points in the objective space. For making tradeoffs in multi-dimensional objective space, the DM is interested in the Pareto optimal solutions. For this, it is advantageous to reduce the dimensionality of the Pareto front to its *true* dimensions. Hence, while using PCA based analysis for dimensionality reduction for multi-objective decision making, it is necessary to consider non-dominated solutions as sample points rather than using random points from the objective space. Depending on the problem and requirements, the PCA based analysis can be used without the iterative step if the sample points are selected from the set of non-dominated solutions.

One of the main advantages of this technique is that it is based on the well known and mathematically robust concepts of PCA. It can be also readily incorporated within a Pareto optimality based MODM framework. The main drawback though is that it does not offer any means of assessing and comparing non-dominated points

obtained before and after the dimensionality reduction. This drawback is addressed by the next technique discussed in the following section.

4.3.1.2 Based on Preserving the Dominance Structure

The PCA based technique does not address how the solution space changes in terms of dominance structure by removing certain objectives. Brockhoff and Zitzler [65] propose a dimensionality reduction technique based on the preservation of the dominance structure. This technique also help address the questions left unanswered by the PCA based technique.

This approach is described here in brief; a complete explanation of the theory behind it is provided by Brockhoff and Zitzler [69, 70]. The authors start with the assumption that the underlying dominance structure is given by the *weak* Pareto dominance relation. It is defined as:

$$\preceq_{\mathcal{F}'} := \{(x, y) \mid x, y \in X \wedge \forall f_i \in \mathcal{F}' : f_i(x) \leq f_i(y)\}$$

where, $\mathcal{F}' \subseteq \mathcal{F} := \{f_1, f_2, \dots, f_m\}$ and X is the set of points in $|\mathcal{F}|$ dimensions. If $(x, y) \in \preceq_{\mathcal{F}'}$ and $x \preceq_{\mathcal{F}'} y$, it is called as x weakly dominates y with respect to the objective set \mathcal{F}' . If neither solution weakly dominates the other, they are said to be *incomparable*. Based on this concept of weak Pareto dominance, the authors have defined a *minimum objective subset* (MOSS) problem. The problem is to find a minimum cardinality subset $\mathcal{F}' \subseteq \mathcal{F}$ such that $x \preceq_{\mathcal{F}'} y \Leftrightarrow x \preceq_{\mathcal{F}} y \forall x, y \in X$. Thus all $\mathcal{F} \setminus \mathcal{F}'$ are considered redundant and can be ignored while preserving the dominance structure of X . This is illustrated in the following example. A parallel coordinate plot is shown in Figure 27 for three randomly selected points x_1 , x_2 , and x_3 from the Pareto front of the knapsack problem of Table 6. The horizontal axis shows five objectives and the vertical axis shows the relative objective value for each point. As observed from the figure, all three points are pairwise incomparable with respect to all five objectives. It can be further observed that V_1 , and V_3 and V_2 , and

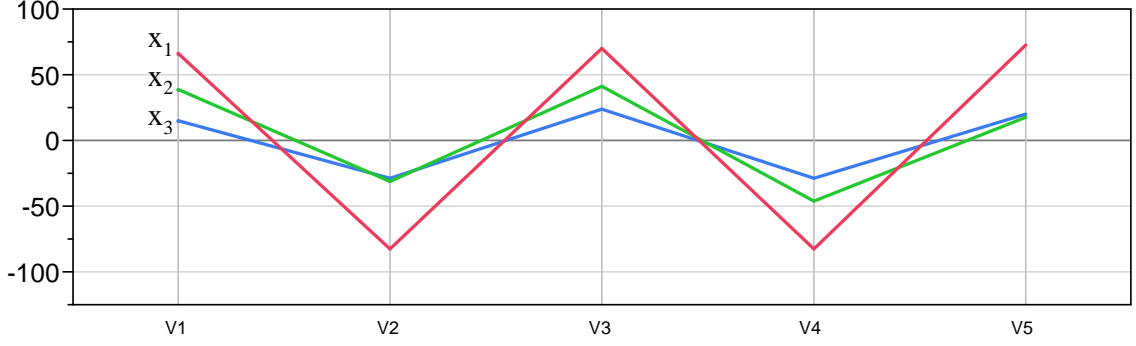


Figure 27: Parallel Coordinate Plot for Three Item Combinations

V_4 indicate redundancy among objectives. The relation $x_3 \preceq_{V_1} x_2 \preceq_{V_1} x_1$ is same as $x_3 \preceq_{V_3} x_2 \preceq_{V_3} x_1$. Similarly, the relations \preceq_{V_2} and \preceq_{V_4} are the same. Thus the objectives V_3 and V_4 can be ignored while preserving the dominance structure of the solutions. With respect to $\mathcal{F}' := \{V_1, V_2, V_5\}$, the three points are still pairwise incomparable.

There are instances where this type of dimensionality reduction while preserving the dominance structure is not be possible. Moreover, the DMs can be interested in reducing the dimensionality even further while accepting some change in the dominance structure. For this purpose, Brockhoff and Zitzler [70] introduce a measure δ to quantify the change in dominance structure due to dimensionality reduction. They further extend the MOSS problem to δ -MOSS to find a subset of objectives with minimum cardinality and maximum δ change in the dominance structure. To understand this concept, lets us consider the previous example in Figure 27. The \mathcal{F}' is further reduced and the new subset $\mathcal{F}'' := \{V_1, V_5\}$. Now the dominance structure changes and $x_3 \preceq_{\mathcal{F}''} x_1$ while $x_3 \not\preceq_{\mathcal{F}} x_1$. For $x_3 \preceq_{\mathcal{F}} x_1$ to hold, the objective values of x_3 have to be lower by $\delta = 55$. This measure δ is used to evaluate the change in the dominance structure induced by a subset of \mathcal{F} . For the subset \mathcal{F}' of current example, there is no change in the structure with respect to \mathcal{F} and hence $\delta = 0$ for \mathcal{F}' . Based on the concept of δ , Brockhoff and Zitzler [70] also introduce a related problem

of minimum objective subset of size k with minimum error (k -EMOSS). Here, the problem is to find a subset $\mathcal{F}' \subseteq \mathcal{F}$ such that $|\mathcal{F}'| \leq k$ and \mathcal{F}' has minimum δ with respect to \mathcal{F} .

The exact and greedy algorithms used for dimensionality reduction in this thesis are from Brockhoff and Zitzler [69, 70]. It has been proved that MOSS and all its generalizations are *NP*-hard; hence, the exact algorithm can take significantly long time for solving large problems. For the example knapsack problem, a greedy algorithm for δ -MOSS is executed with $\delta = 0$ to determine if there are any redundant objectives. The entire Pareto optimal set with 4,005 points is used for this analysis and it is observed that no dimensionality reduction is possible without altering the dominance structure. An exact algorithm for k -EMOSS formulation is implemented on the same set of points to investigate the possibility of reducing dimensions by accepting some error in the dominance structure. Here, $k = 4$ and the algorithm searches for the subsets of objectives with cardinality less than 4 and δ minimum; the result is listed in Table 10. The algorithm took just over two hours to run on a Pentium 4 2 GHz machine. The second column of Table 10 lists the cardinality of the corresponding subset, third column lists the objectives present in the subset and the last column has the corresponding δ value. This table can be used to select the objectives of interest based on the error one is willing to accept. For example, if one is interested in at most four objectives, then set number seventeen with $\mathcal{F}' := \{V_2, V_3, V_4, V_5\}$ has the lowest δ value of all subsets with cardinality four. If one is willing to accept more error, set number 8 or 15 can be used with only three objectives. Moreover, subsets can be selected based on the preferences of objectives. For example, if one is more comfortable making decisions based on objective V_3 than V_1 , then set number 15 can be selected instead of 8 without any change in the dominance structure (because $\delta = 41$ for both).

Table 10: k -EMOSS Results for $k = 4$

No.	$ \mathcal{F}' $	\mathcal{F}'	δ
1	1	1	103
2	2	1, 2	46
3	3	1, 2, 3	44
4	3	1, 2, 5	42
5	4	1, 2, 4, 5	27
6	3	1, 3, 4	43
7	2	1, 4	46
8	3	1, 4, 5	41
9	1	2	97
10	2	2, 3	47
11	3	2, 3, 5	42
12	2	2, 5	54
13	1	3	103
14	2	3, 4	47
15	3	3, 4, 5	41
16	1	4	97
17	4	2, 3, 4, 5	21
18	2	4, 5	54
19	1	5	103

The advantage of using this technique lies in the fact that it attempts to preserve the dominance structure of the solution space. The preservation of dominance structure, that is $\delta = 0$ for an objective subset, indicates that the dimensionality of the Pareto front is preserved. This property is most useful to the DMs when they are making tradeoffs along the Pareto frontier as there is no loss of information even with reduced objectives. Moreover, the ability to measure the change in dominance structure imparts flexibility to this technique. It eliminates the main drawback of the PCA based technique. Now, the objectives can be reduced while being aware of the extent of change in solution structure. The main drawback of this technique is that it is computationally expensive and hence cannot be integrated within the Pareto search algorithms.

Table 11: Comparing PCA Based and Dominance Based Techniques

PCA based	δ based	δ
1, 4, 5	1, 4, 5	41
3, 4, 5	3, 4, 5	41
1, 3, 4, 5	N/A	41
N/A	2, 3, 4, 5	21

4.3.1.3 Comparing Two Techniques

It is interesting to compare the PCA and dominance based techniques for dimensionality reduction. The comparison is based on the complete set of Pareto optimal solutions with 4,005 points for the example knapsack problem. It can be recalled that implementation of PCA based technique with TC of 95% on the Pareto set resulted in the selection of three objectives $\mathcal{F}'_1 := \{V_1, V_4, V_5\}$. As observed from Table 11, the same set is optimal for k -EMOSS problem with $k = 3$ and the corresponding error is $\delta = 41$. All δ values in third column of the table are calculated with respect to the set \mathcal{F} of all objectives. If, in the final step of PCA based technique V_3 was selected in place of V_1 , the resultant objective subset $\mathcal{F}'_2 := \{V_3, V_4, V_5\}$ is still optimal and the dominance structure is similar to \mathcal{F}'_1 ($\delta = 41$ for both). Now, what would be the advantage, if any, of retaining both V_1 and V_3 and selecting four objectives $\mathcal{F}'_3 := \{V_1, V_3, V_4, V_5\}$? As it is observed from the table, this is not an optimal set for $k = 4$ and there is no gain in having both V_1 and V_3 together in \mathcal{F}'_3 . The dominance structure is similar to \mathcal{F}'_1 and \mathcal{F}'_2 with $\delta = 41$. Thus, one of the objective among V_1 and V_3 is clearly redundant in \mathcal{F}'_3 . This behavior is suggested by the PCA based technique but proved by the dominance based technique. When one is interested in just eliminating one objective from \mathcal{F} , the dominance based technique gives the best answer with $\mathcal{F}'_4 := \{V_2, V_3, V_4, V_5\}$ and $\delta = 21$. This set is not obtained using PCA based technique.

From the above study and based on the assumption of similarity of knapsack and technology selection problem, it can be stated that:

Hypothesis: Dimensionality of the Pareto hyper-surface in a multi-objective technology selection problem will be smaller than the number of objectives.

Supporting Experimentation: Plausibility of this hypothesis will be checked using the dominance structure based dimensionality reduction technique implemented on a subset of Pareto optimal solutions.

4.3.2 Search for Pareto Optimal Solutions

Searching for a representative subset of the Pareto optimal solution set is the most significant challenge with *a posteriori* preference articulation framework. There are many approaches available in the literature for Pareto optimization. The most common of the approaches is by using the weighted sum method as described in Section 4.1.1 and illustrated in Figure 20. Here, the weights are parametrically varied for each objective and the problem optimized for given weights vector. Any optimizer can be used for this application, for example, Roth et al. [71] use genetic algorithm as a point optimizer for iteratively varied weight vector. In addition to the significant drawbacks described in Section 4.1.1.1, this technique can be computationally very expensive for higher dimensional problems.

To eliminate some of the limitations of weighted sum based technique for Pareto optimization, Das and Dennis [72] propose Normal Boundary Intersection (NBI) technique. The process is carried out by first defining a Convex Hull of Individual Minima (CHIM). This is a line, surface, or hypersurface formed by connecting the extreme points in two, three, or more objectives respectively. These extreme points are the most optimal points for each objective when considered independently. For a two dimensional problem illustrated in Figure 28, CHIM is represented by segment \overline{ab} . The Pareto optimal point is the intersection point of the normal coming from a point on CHIM towards the origin (for minimization problems) and the boundary of the

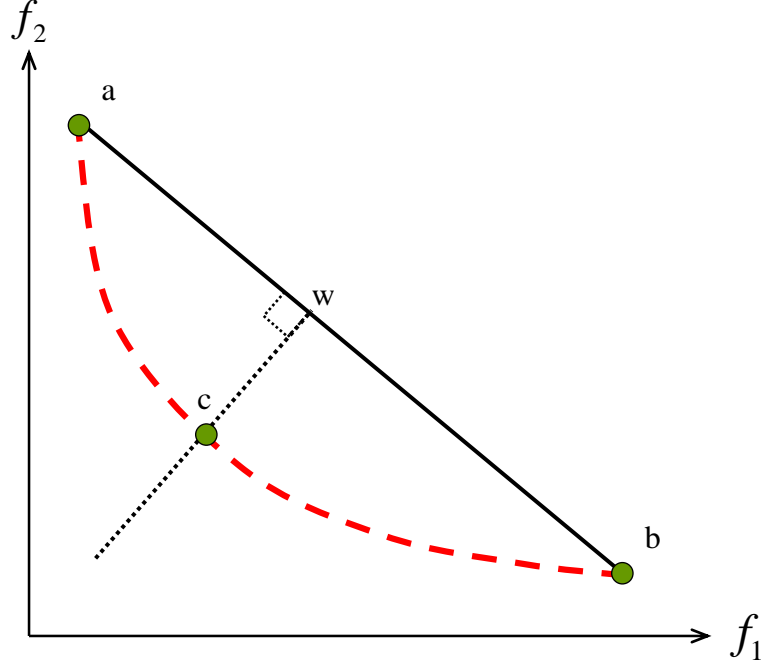


Figure 28: Normal Boundary Intersection

objective space. This is shown in Figure 28 with arc (acb) as the boundary of the solution space, \vec{wc} as normal emanating from point w on CHIM. The point w represents a certain weight vector. By using different user defined weight vectors, Pareto optimal points can be obtained along the front. This technique produces evenly spaced Pareto points given evenly distributed set of weights, irrespective to the variation in scales of different objectives. Though NBI can be extended for multiple objectives, the computational efficiency reduces with increasing objectives because each point on the front has to be individually optimized.

As an alternative to the weighted sum based techniques, ϵ -constraint technique can be also used to identify the Pareto optimal points. This technique is implemented by Cheng et al. [73] for optimization in two dimensions of profit and risk. In this technique, one of the objective is selected to be optimized and others are converted into constraints by setting bounds on them. All points on the Pareto frontier can be generated by successively tightening the constraints. Another noteworthy method

for finding Pareto optimal solutions is based on the homotopy curve tracking technique. Rakowska et al. [74] demonstrate this method on a two dimensional problem of optimizing control and structural objectives simultaneously. They use a homotopy algorithm developed by Chakraborty et al. [75] to trace the Pareto frontier in the objective space. Though these methods are appropriate for two dimensional problems, they become computationally expensive as the dimensions increase.

Techniques based on Evolutionary Algorithms (EAs) are very promising for Pareto optimization. They have been developed and matured in the recent years and work well for large dimensional problems. These are population based techniques and attempt to search for all the points of a representative Pareto set in parallel. They require no gradient information and work exceptionally well for discontinuous and combinatorial problems. Many of these algorithms are based on the concepts of non-domination and niching described by Goldberg [76]. A good introduction to these algorithms and techniques is provided by Coello [63, 77] and Veldhuizen [56, 78] among many others.

Because of the importance of Pareto optimization for solving the technology problem and the intricacies involved with the EA approach, the next chapter is devoted towards the discussion of EAs and their application for Pareto optimization.

4.4 *Summary*

The primary intent of this chapter has been to investigate various MODM techniques and down-select the most appropriate one considering the goals of this thesis. Distinction has been made between the three main classes of MODM techniques: *a priori*, *progressive*, and *a posteriori* preference articulation. There are serious limitations with the *a priori* preference articulation methods. These include, but are not limited to, their inability to find points in the non-convex part of the Pareto front and the difficulty to predict weights that would result in optimized points in the region of

interest. The *progressive* preference articulation techniques are considered very time consuming and inefficient for the decision makers. Finally, *a posteriori* preference articulation class of techniques is considered appropriate for the technology selection problem. In this framework, a subset of technology combinations representing the Pareto front in multi-dimensional objective space will be searched. This set will then be presented to the decision makers to carry out tradeoffs among objectives and select a satisfying technology combination.

There are challenges involved in implementing *a posteriori* preference articulation framework for the complex and multi-dimensional technology selection problem. The first is associated with redundant dimensions present in the problem. *PCA* based and *Dominance* based techniques are shown to be useful to address this challenge. It has been demonstrated with a benchmark knapsack problem that dimensionality reduction is possible if one is willing to accept some error in the dominance structure. It is hypothesized that the technology selection problem will also have some redundant dimensions; the dominance based technique can be used to check this and also reduce the dimensionality of the problem. Other significant challenge is to search for a representative subset of Pareto optimal points. Classical weighted sum technique and also the Normal Boundary Intersection technique are deemed inappropriate for the task. Evolutionary algorithms seem to provide notable possibilities for Pareto optimization. This is further explored in the next chapter.

CHAPTER V

EVOLUTIONARY ALGORITHMS FOR PARETO OPTIMIZATION

It is understood from the previous discussion that at the highest level, the technology selection problem is a multi-objective decision making problem. Multi-objective optimization is a crucial part of the process. In the previous chapter it was shown that *a posteriori* preference articulation with Pareto optimization is appropriate for the technology problem. The focus of this chapter is towards addressing the question: *How to efficiently search for non-dominated points in multiple dimensions?*

This chapter will explore the use of Evolutionary Algorithms (EAs) for the purpose of Pareto optimization. A brief introduction to EAs is provided in the initial section. Main issues faced while applying EAs for Pareto optimization are discussed next, followed by introduction to some of the most popular algorithms for the task. The promising algorithms are compared using a benchmark knapsack problem and the best one is selected for the technology selection problem. To investigate the efficacy of the selected algorithm, its results are compared with results from a random search.

5.1 Evolutionary Computation

Evolutionary computation is the study of computational systems that use inspiration from the natural process of evolution and adaptation. The main areas included in the study of evolutionary computation are evolutionary programming, evolution strategies, genetic algorithms and genetic programming. *Evolutionary Algorithms (EAs)* is the general term used to include the first three areas. Spears et al. [79] and Yao [80] give a comprehensive description of the similarities and subtle differences between

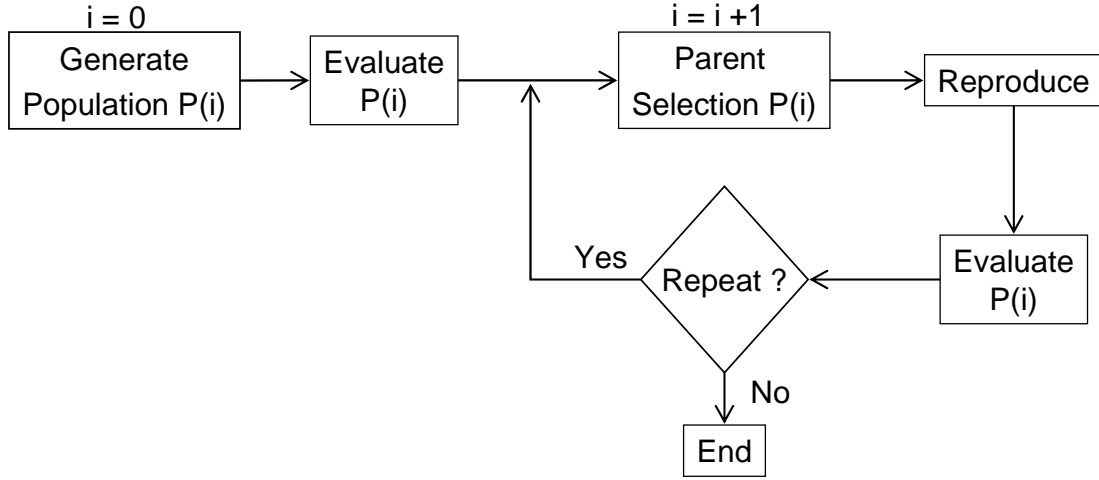


Figure 29: General Outline of EA

different types of EAs. Whitley [81] provides a good description of EAs and some of the intricacies involved, such as schema theorem, representations, etc. An outline of a typical evolutionary algorithm is illustrated in Figure 29.

The following subsection details some of the primary reasons behind the decision to use an EA for the technology selection problem. Later, a brief introduction to *No Free Lunch* theorems is provided and their implications are discussed.

5.1.1 Why Evolutionary Algorithms?

Technology selection has been shown to be an NP-hard problem. Evolutionary algorithms are particularly well suited for such problems [66, 82]. The main reason for this is the inherent parallelism of the techniques i.e. they process a set or a population of solutions simultaneously. Apart from the empirical evidence about the suitability of this approach for various theoretical [83, 84, 85] and practical [86, 87, 88] problems, there are some particular characteristics (described below) of the technology problem that make EAs a good choice.

Combinatorial: The concerned problem is a *Boolean* combinatorial problem. Here

the technologies can either be selected (1) or not selected (0). Therefore each bit on the binary chromosome string represents an actual technology and not its encoding. The operations of EA such as crossover, mutation, etc. take place in the actual technology space or phenotype space. Thus the information is conserved and transferred in a true *building block* sense [76]. It has also been argued by Radcliffe [89] that EAs are more efficient when the genetic operators are defined in the phenotype space rather than genotype space¹.

Multi-Dimensional: Lower dimensional problems can often be solved more efficiently by traditional techniques of mathematical programming [90]. Chu and Beasley [91] demonstrated that a heuristic based GA can be used efficiently compared to other techniques to solve a multi-dimensional knapsack problem. Our focus is towards a multi-dimensional problem and EAs can be effectively used for this purpose. Moreover, EAs are known to be very efficient for generating a subset of Pareto optimal solutions.

Constrained: EAs are inherently unconstrained search methods. It is necessary to devise different techniques to incorporate constraints in these algorithms. Michalewicz [92] and Ceollo [93, 94] provide a comprehensive survey of wide variety of constraint handling techniques used over the years. Constraints in the technology selection problem arise from different types of relations among the technologies as explained in previous chapters. The information about these relations can be used to devise a heuristic based operator to maintain feasible population.

Discontinuous: Genetic algorithms and other EA methods are fundamentally discrete variable methods. These metaheuristics methods are also ideal for discontinuous objective spaces as they do not rely on the gradient information. Even

¹encoding or representation space

though gradient based methods quickly converge to optimal solution, they are not efficient in non-differentiable or discontinues problems. For this case, the parameters are discrete as well as the objective space can be discontinuous depending on the type of system model used.

Epistatic: Epistasis is the degree of interaction among parameters (technologies in this case) as manifested in an objective function. If there is no epistasis, then the mapping of parameters on the objective is linearly independent. In such cases the parameters can be optimized independently and a simple algorithms like hill-climbing will outperform any advanced EA. On the other hand, if there is unbounded epistasis, i.e. the contribution of all the parameters depends on the values of all others, the problem is extremely difficult to handle by EA or any other methods. It has been suggested that EAs and other metaheuristics excel in searching problems with bounded epistasis [95].

5.1.2 No Free Lunch Theorems

No discussion about the applicability of EAs can be complete without the mention of theoretical work attempted in the recent past demonstrating the limitations of stochastic search algorithms. Some of the most important results of these studies can be found in the seminal work of Wolpert and Mcready [96, 97] called *No Free Lunch* (NFL) theorems for search and optimization. Radcliffe and Surry [95], and English [98] expand some of the results of NFL and explore the ramifications of these theorems on search and optimization. Culberson [99] gives a good informal explanation about NFL theorems especially in the light of complexity theory and investigates its implications on evolutionary computing.

The NFL theorems prove that all algorithms that search for an extremum of a function perform exactly the same when averaged over all possible functions. It states that if an algorithm *A* outperforms algorithm *B* on some objective functions, then

B must outperform A on others. That is, all algorithms, even a random search, will perform the same on average on all the search spaces. The immediate consequence of this result is that it proves the futility of trying to devise a general purpose algorithm that can efficiently search any objective space. A general purpose algorithm may be devised but it will be akin to a Swiss army knife, as English [98] compares it, able to do many jobs, but none particularly well.

5.1.2.1 Implications of NFL theorems on EAs

Instead of discouraging the evolutionary computation community, the NFL theorems provide direction for the improvements in EAs and other metaheuristic search algorithms. These algorithms are occasionally promoted as a *cure for all* optimization problems but NFL theorems put some limitations on such claims. NFL theorem implies that the best ways to devise an efficient search algorithm and to know that it will be efficient, *before* trying it out, is to tailor it according to the the problem. That is, to use some problem specific information or structure that is known and exploitable, and reflect this in the algorithm selected; only then one can prefer one algorithm over another. Otherwise there can be no basis for selection of the algorithm and no formal assurance that it will be effective.

Representation schemes and operators have a prominent role in successful implementation of EAs. These methods can be made more effective by incorporating domain specific knowledge into representation and operators used [95]. One can thus trade performance increase in the domain of interest with performance decrease in the other domains.

5.2 Pareto Optimization Using EAs

The goal of Pareto optimization is to obtain a set of points that approximates the Pareto surface in the objective space, and their corresponding parameter values. The need for Pareto optimization for a high dimensional space is one of the main reason

behind selecting evolutionary algorithms. As mentioned before, the advantage of using EA's for Pareto optimization is their ability to work with a large population of points simultaneously. These population based algorithms exploit the knowledge of the entire population to drive the search towards Pareto surface in all directions.

As Zitzler [67] states, approximating the Pareto surface is in itself multiobjective task; first, one has to reduce the distance between the actual and approximate surface and the second is to ensure even distribution of points on the surface. These tasks lead to few questions that have to be answered while designing the algorithm. The question of assigning a scalar fitness value to a point in multiple dimensions has to be addressed to compare the individuals within the population to accomplish the first task. For the second task, a mechanism has to be devised so that the final set is distributed evenly over the Pareto surface. Three main issues that have to be addressed by the Pareto optimization algorithms are *fitness assignment*, *distribution along the surface* and *elitism*.

5.2.1 Fitness Assignment

This can be considered as the most important function of the algorithm. Fitness assignments to the individuals in a population ensures the gradual movement of approximate surface towards the actual Pareto surface through the generations. There are three main schemes for fitness assignments:

Criterion based fitness assignment considers single objective at a time. Schaffer [100] in his seminal work on evolutionary multiobjective optimization called vector evaluated genetic algorithm (VEGA) used this scheme for fitness assignment. It uses the objectives in equal proportion to calculate fitness of individuals in the population. That is, if there are n objectives and k individuals in the population, then k/n portion of population will use one objective and the same number of individuals will use another objective. Kursawe [101] suggested

using a user defined vector that gives the probability of each objective to be considered as the fitness criterion. This vector can also be allowed to change over time.

Scalerizing the objectives using a parameterized function. This approach is based on the traditional multi-objective optimization technique. The advantage of this method is that once the objectives are parameterized in a single function and this is used for fitness assignment, standard selection criteria can be used without any modification. For most instances, the parameters or weights are assigned randomly at each step. But, as described in the previous chapter, this technique has serious limitations.

Non-dominance based strategy calculates the fitness of an individual based on Pareto dominance. It was first introduced by Goldberg [76] in 1989 and many derivatives have been developed since. This scheme gives highest fitness values to non-dominated individuals and progressively lower fitness values to dominated points. This is the most successful technique used for Pareto optimization using EA; many algorithms such as Non-dominated Sorting GA, Niche Pareto GA, Fonseca and Fleming GA, etc. use this technique; they will be explained in detail in the later sections.

5.2.2 Distribution Along the Surface

Depending on the type of fitness assignment used, there can be two main types of techniques to obtain diversity along the frontier. If the fitness assignment is criterion based or a scalerized objective is used, the criterion or weight vector is changed over time. Usually the GA or other EA is iteratively run with different weight vector, each time optimizing a particular region of the Pareto surface as defined by the vector. The change of criterion or weight vector can be random or in predefined steps. Though simple to implement, this technique may not be able to find an evenly distributed

population along the frontier. It also tends to be computationally intensive especially for large number of objectives. The technique also does not take the advantages offered by a population based search.

Niching is the other technique employed in Pareto based MOEAs. It is based on natural mechanism of formation of distinct species exploiting different niches in the ecosystem. In EAs, niching amounts to formation of different subpopulations, each optimizing a specific region of the Pareto surface. Horn [59] gives a detailed explanation of the philosophy behind niching as employed for Pareto optimization. Niching is basically a density based technique where fitness sharing is employed depending on the density of individuals within ones neighborhood. That is, if there are many individuals in ones neighborhood, the chances of it getting selected decreases. There are various techniques for density estimation and a brief introduction to these is provided by Zitzler [67]. Niching is one of the most well known techniques of diversity preservation and is used in most of the Pareto based GA.

5.2.3 Elitism

Elitism, in terms of evolutionary computation refers to the strategy of retaining more fit individuals of a population from one generation to the next. This allows them to take part in the evolutionary operations more than once and prevents loss of good solutions due to random effects of evolutionary process. There are various implementations of elitism, starting from retaining the best individual of a generation in basic GA to more sophisticated techniques of maintaining an auxiliary population. While maintaining an auxiliary population, or archiving, as its commonly known, most fit individuals of population are copied at every generations. This archive can just be used as a storage or can be integrated with the EA where some individuals of online population are replaced by the individuals from the archive. As compared to single objective optimization, the incorporation of elitism in multi-objective optimization

is more complex and most MOEA use some combination of dominance and density criteria to decide which individuals are to be included in the archive. Some of the well known algorithms that use elitism are NSGA-II, SPEA, PAES, etc. Knowles [90] provides a detailed survey about the history of elitism used in evolutionary computation and different techniques used in MOEAs.

With the basic knowledge of Pareto optimization using EAs, the following sections detail some of the algorithms and techniques used for this purpose. Many surveys have been published by prominent researchers in the field of Evolutionary Multi-Objective Optimization (EMOO) describing [102, 63] and comparing [103, 66] various techniques. To make this document comprehensive, following sections discuss three main methods that use non-domination and sharing techniques suggested by Goldberg [76]. One of the recently developed methods that use elitism or archiving is also discussed.

5.3 Fonseca and Fleming GA (FFGA)

FFGA as the name suggests was proposed by Fonseca and Fleming [104] in 1993 and is called as Multi Objective GA (MOGA) by them. They suggest using a non-dominated rank based fitness assignment. Here, when an individual i of generation t is dominated by ρ_i^t individuals, its rank is given by:

$$rank(i, t) = 1 + \rho_i^t$$

As a result of this ranking scheme, not all ranks will necessarily be represented in the population; as Figure 30 illustrates, rank 2 is absent. After the individuals are ranked and sorted, the fitness is assigned to them by interpolating from the best (i.e. ranked 1) to worst individuals, according to some linear or non-linear function. The fitness of individuals with same rank is averaged so that all of them will be sampled at the same rate.

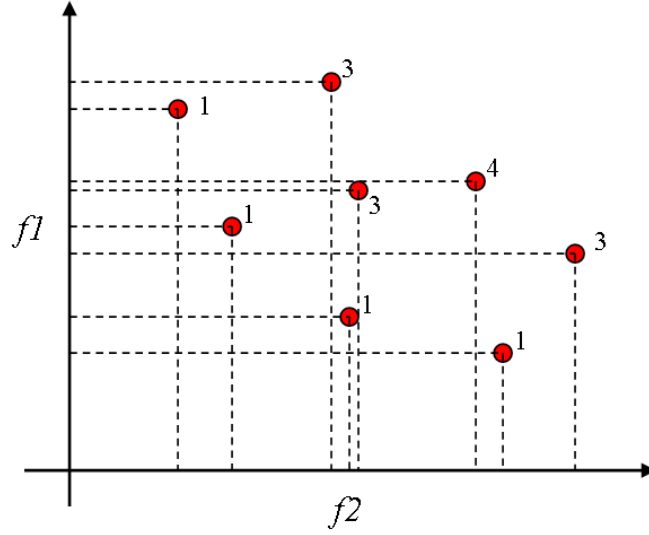


Figure 30: Rank Assignment in FFGA

Niche formation method for fitness sharing is employed in MOGA to prevent genetic drift. The fitness sharing is implemented in the phenotypic space to obtain a well distributed solution in the objective space. This introduces a parameter called niche size σ_{area} which needs to be set carefully and the authors provide a theory for estimating it according to the properties of the problem. The parameter σ_{share} determines the distance between two individuals on the Pareto surface. The fitness of individuals is reduced if the distance between them is less than σ_{share} .

This method also includes a higher level DM in the optimization process. The aim is to reduce the size of the solution set and zoom in a particular area of the Pareto surface that is of more interest to the DM. The multi objective ranking method for fitness assignment is modified to include the goal information provided by the DM. This method falls under the category of *progressive* techniques for articulation of DM's preferences as mentioned in the previous chapters.

5.3.1 Advantages and Shortcomings

The simplicity of implementing this method is the main advantage it provides. The other is the inclusion of DM in the loop who can influence the direction of search towards more interesting areas.

The main weakness is the need for accurate setting of σ_{share} which has large impact on performance of the method. The inclusion of DM in optimization loop may not be desired in some applications.

5.4 *Non-dominated Sorting GA I & II (NSGA I & II)*

Srinivas and Deb [105] proposed non-dominated sorting GA(NSGA) in 1994. In this method a ranking selection technique is used to select good points and a niching method used to maintain a stable subpopulation of good points. The basic outline of the method is illustrated in Figure 31.

NSGA is different from other methods in the way it implements the fitness assignment for selection. Initially, all the non-dominated points in the population are identified and assigned a dummy fitness value. All these points have the same fitness now. After this, sharing is implemented in this non-dominated set of points. For this, dummy fitness of each point in this set is reduced by some value, obtained by dividing the fitness by a quantity proportional to the number of neighbors of that particular individual. For the next step, this set of non-dominated individuals is ignored and the process repeated. The dummy fitness of the new set is kept below the minimum shared fitness value of the previous non-dominated set. The process continues until entire population is sorted into various fronts. The population is reproduced using proportionate selection. The crossover and mutation operators are implemented as usual and the process continued for required generations.

Because of the drawback of high computational complexity and the need to specify sharing parameter, Deb et al. [106] proposed an improved version of NSGA called

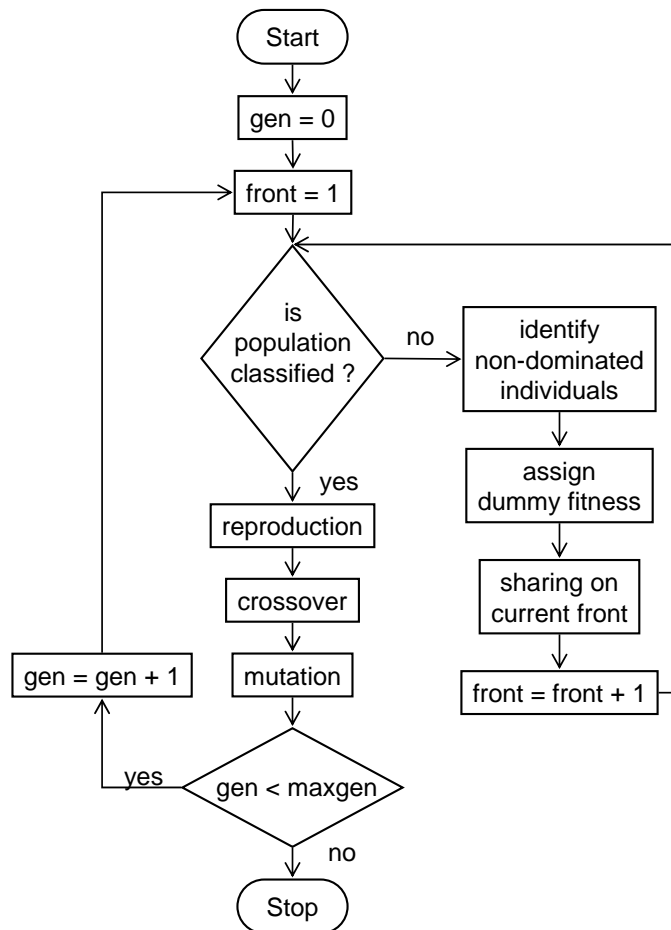


Figure 31: Non-dominated Sorting GA

NSGA-II in 2000. This new algorithm uses a different and more structured technique to compare and identify non-dominated fronts that reduces the overall complexity of the method. Sharing is performed by comparing a quantity called *crowding distance* of the individuals; this eliminates the need for specifying a sharing parameter. Crowding distance is the average distance of two points on either side of the individual along each axis. The main GA loop of NSGA-II also implements elitism by comparing current population with previously found non-dominated set of points.

5.4.1 Advantages and Shortcomings

As mentioned before NSGA is computationally intensive compared to other methods and the results are very sensitive to the sharing parameter. NSGA-II solves most of drawbacks of NSGA. Deb et al. [106], using different benchmark problems, favorably compare this method with two other methods that also use elitism.

The only drawback with NSGA-II approach is that one has to be careful while coding the algorithm as the implementation is little complicated when compared to other methods. Moreover, this algorithm loses its effectiveness when the problem dimension is large.

5.5 *Niched Pareto Genetic Algorithm (NPGA)*

NPGA as proposed by Horn et al. [107] is designed along the natural analogy of evolution of distinct species exploiting different *niches* or resources in the environment. A canonical GA is purely competitive where the best individuals quickly takeover the population. Whereas, when niching is included in the GA scheme, the populations tend to cooperate and the final set converges to a population of diverse species that are distributed along the Pareto frontier. The philosophy behind niching and NPGA has been discussed in detail by Horn [59].

The basic implementation of NPGA concerns modifying the selection function of GA. One of the most widely used selection technique is *tournament* selection. Here,

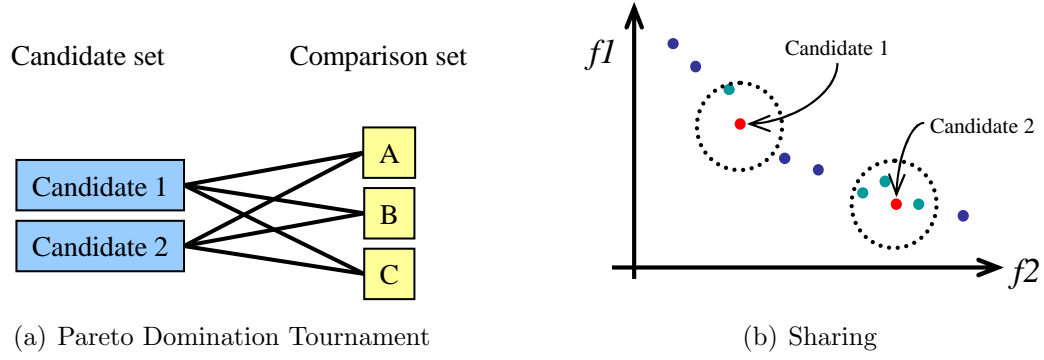


Figure 32: NPGA Selection Operators

a subset of the population is randomly chosen and the best candidate in this set is selected. This implementation assumes that a single answer to the problem is desired. For NPGA, the selection method is modified to have multiple answers to the multi-objective problem. The selection method includes two main components: *Pareto-domination tournaments*, and *Sharing*.

5.5.1 Pareto-Domination Tournaments

The tournament selection is altered to use multiple attributes for creating a Pareto frontier. To increase the domination pressure in the tournament selection, two *candidates* for selection and a comparison set of individuals is picked at random from the population. A graphical illustration of this type of tournament is provided in Figure 32(a). The number of individuals in the comparison set can be adjusted according to the requirements of the domination pressure. Each of the candidates is compared against each of the individuals in the comparison set. If one candidate is completely dominated by the comparison set and the other is not, then the latter is selected. If both the candidates are dominated or non-dominated, then sharing is used to decide the winner.

5.5.2 Sharing

Sharing helps to choose candidates when there is a tie after the tournament. If one of the candidates is randomly selected then, genetic drift will cause the population to group around in a single section of the Pareto front. To prevent this, equivalence class sharing in the objective space is implemented. Here, no preference is given to the two individuals regarding their objective values as they are already in the same equivalence class after the tournament. They are selected on the basis of density of population points in the neighborhood of a particular candidate. This density is calculated in the form of *niche count*, that is, the number of individuals present within the *niche radius* σ_{sh} of a particular candidate. This is illustrated in Figure 32(b) where the radius of circle represents σ_{sh} . The niche radius determines how far apart the individuals lie on the final Pareto frontier. The value of σ_{sh} is under the control of the user and can be changed according to the requirements of a given problem. In order to determine σ_{sh} , Horn et al. [107] suggest dividing the total surface area of the Pareto frontier with population size:

$$\sigma_{sh} \approx \frac{A_{pareto}}{N}$$

In this case, ideally, the population N will be equally distributed, with σ_{sh} units apart from one another, across the Pareto front. As for A_{pareto} , one may not know the exact area of the front but it is possible to determine the ranges of objective functions and with that, the range of A_{pareto} . With M and m denoting the vector of maximum and minimum magnitudes of objective functions respectively, for a two dimensional problem, A_{pareto} will be greater than the hypotenuse given by:

$$A_{pareto} > A_{min} = \sqrt{(M_1 - m_1)^2 + (M_2 - m_2)^2}$$

The sum of the objective value ranges determines the upper bound for A_{pareto} :

$$A_{pareto} < A_{max} = (M_1 - m_1) + (M_2 - m_2)$$

In general, A_{max} will be the sum of all the faces of a hyperparallelogram of edges $(M - m)$ determined by Equation 16 [104].

$$A_{pareto} < A_{max} = \sum_{i=1}^n \prod_{\substack{j=1 \\ j \neq i}}^n (M_j - m_j) \quad (16)$$

It has been noticed that large difference in the magnitudes of various objectives can affect the distribution of population along the Pareto front [59]. This is due to the fact that Euclidian distance is used to measure the separation of two points on the Pareto frontier in n dimensional objective space. This metric does not differentiate between the ranges and magnitudes of objective space. Hence one can have a skewed Pareto front if the objective values are used in the raw form. One of the most straightforward ways to avoid this kind of niching bias is to scale the objectives so that they are at the same magnitude.

5.5.3 Advantages and Shortcomings

This method does not require the entire population to be ranked according to non-domination. As a consequence it is faster than FFGA and NSGA [63]. The implementation of NPGA is considerably straightforward by changing the reproduction operator of a canonical GA.

The main weakness include the requirement of scaling the objective values and the presence of an extra parameter of tournament size. The results are considerably dependent on the values of niche radius and tournament size. Moreover, there is a good possibility of losing good solutions as the method does not use elitism.

5.6 *Strength Pareto Evolutionary Algorithm I & II (SPEA I & II)*

SPEA developed by Zitzler and Thiele [66] combines some of the proven and new techniques to find a subset of Pareto optimal solutions. The distinguishing factor of this method is the existence of a secondary or external population that stores

all the non-dominated solutions so far. These external individuals also participate in the selection process. Scalar fitness values are assigned according to the Pareto dominance of individuals in relation to the non-dominated solutions stored in the external population only; the fitness value is called *strength* of the individual. A Pareto based niching technique is used to distribute the individuals equally along the front; this technique does not require the sharing or niching parameter. When the number of solutions in the external population increase above a specified limit, clustering is employed. This reduces the size of external population without destroying the characteristics of Pareto hypersurface.

Zitzle et al. [108, 109] have introduced an improved version of SPEA called SPEA2. A flow chart for SPEA2 is shown in Figure 33. The new algorithm attempts to eliminate the weaknesses of its predecessor by incorporating new knowledge gained in the field of evolutionary multiobjective optimization (EMO). The main changes in the new algorithm include improved fitness assignment, nearest neighbor density estimation and a new archive truncation method. The strength of a certain individual is assigned by the number of dominators in both the population and archive. For each individual i in the population P_t and archive \bar{P}_t of generation t , the strength is given by Equation 17.

$$S_i = | \{j | j \in P_t + \bar{P}_t \wedge i \succ j\} | \quad (17)$$

The raw fitness of the individual is calculated on the basis of the strength of dominators in both populations and is given by Equation 18.

$$R_i = \sum_{j \in P_t + \bar{P}_t, j \succ i} S_j \quad (18)$$

An even distribution along the Pareto surface is achieved by including density information in the final fitness value. An adaptation of k^{th} nearest neighbor method is used for estimating the density. The technique uses inverse of the distance to the k^{th} nearest neighbor as density estimate D_i . The final fitness F_i , given by Equation 19, is

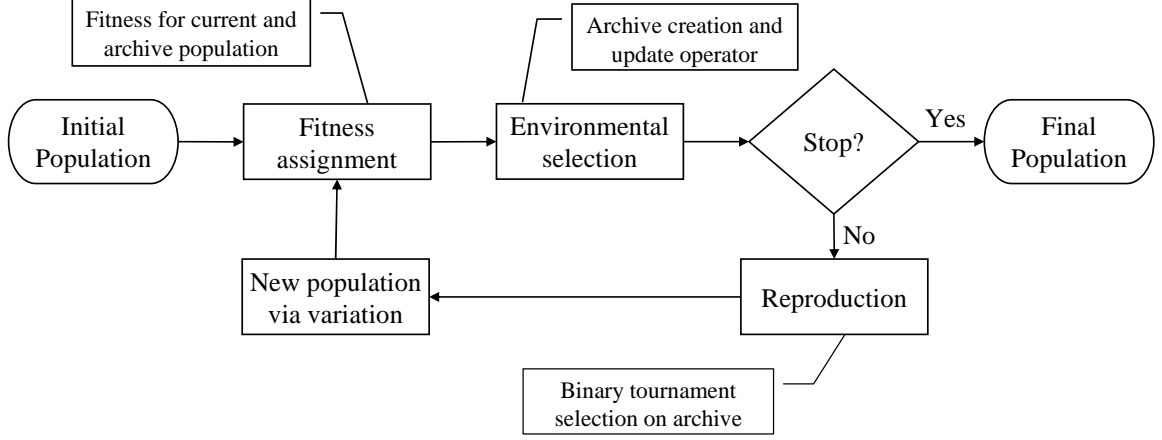


Figure 33: Strength Pareto Evolutionary Algorithm II

obtained by adding the raw fitness and density estimate and is to be reduced through the generations.

$$F_i = R_i + D_i \quad (19)$$

The external archive of SPEA2 is of fixed size \bar{N} . All non-dominated individuals, i.e. individuals with fitness less than 1, in $P_t + \bar{P}_t$ are copied to \bar{P}_{t+1} . If this number is less than \bar{N} , dominated individuals with fitness values in the lower spectrum are copied to the archive. In case the number of non-dominated individuals is more than \bar{N} , the archive is truncated by iteratively removing the individuals until $|\bar{P}_{t+1}| = \bar{N}$. The k^{th} nearest neighbor criteria is used for this purpose. The individual with minimum distance to the k^{th} neighbor is purged at every iteration.

5.6.1 Advantages and Shortcomings

SPEA and SPEA2 do not require any distance parameter or tournament size that can have considerable effect on the quality of solution. SPEA combines some of the well established strategies in EMO field. According to Zitzler et al. [66, 103] SPEA compares favorably with other EMO methods. Given the presence of external population and implementation of elitism, these method have good possibility of obtaining a well distributed frontier.

The main drawback of these algorithms, especially SPEA2, is that the computational complexity can be considerably higher if the size of population and archive is high. Moreover, as the algorithm scans the entire population for non-dominance, computational complexity can rapidly increase with increase in objective functions. Care has to be taken while implementing the algorithm, distance measurement and comparisons have to be done in systematic order so as not to increase the computational complexity.

From the above discussions, two algorithms look promising: NPGA – because of its speed and simplicity, and SPEA2 – because of its accuracy. Moreover, Zitzler et al. [109] have demonstrated the advantages of SPEA2 over other methods for higher dimensional problems. Based on this discussion the following hypothesis is proposed addressing the research question: *How to efficiently search for non-dominated points in multiple dimensions?*

Hypothesis: Pareto optimization of the technology selection problem can be most efficiently accomplished by the Strength Pareto Evolutionary Algorithm II.

Supporting Experimentation: Plausibility of this hypothesis is checked by comparing NPGA and SPEA2 on the benchmark knapsack problem. Efficacy is checked by comparing the SPEA2 results with results from a random search.

The following sections attempt to check the plausibility of the above hypothesis by comparing the two algorithms using a benchmark knapsack problem.

5.7 Comparing NPGA and SPEA II

The performance comparison of NPGA and SPEA2 is carried out using the benchmark problem defined in Chapter 4 in Table 6. It is a 16 item, 5 objective knapsack problem with the aim of minimizing all the objectives. The total number of possible

solution combinations are $2^{16} = 65,536$ out of which, 4,005 are the non-dominated combinations.

5.7.1 Criteria for Performance Comparison

Performance metrics are required while quantitatively comparing two EAs for optimization. These metrics are quite simple when optimizing a single objective. This generally involves observing the convergence behavior of EAs and determining the best solution achieved by each. On the other hand, the performance metrics are significantly complex for Pareto optimizing EAs. Here, one has to qualitatively determine the spread and distribution of Pareto front obtained by the EA, and determine how close is the obtained Pareto front to the *actual* Pareto front. Moreover, to observe the convergence behavior of the EA, an appropriate convergence criteria has to be defined. Zitzler et al. [103] provide one of the most comprehensive discussion in this area of performance comparison of Pareto optimizing EAs.

For selecting the best algorithm, it is reasonable to first check if the algorithm provides a front that accurately represents the actual Pareto front. Let Ω be the set of all Pareto optimal solutions, that is, Ω is the actual Pareto front. The Ω for the benchmark problem is already known and its cardinality is $|\Omega| = 4,005$. Now, let ω_A be the set of non-dominated solutions provided by algorithm A . If the algorithm is working as expected, then ideally $\omega_A \subset \Omega$. But in practice, this is not always the case. ω_A will have some dominated solutions with respect to Ω . These solutions are not part of Ω . Considering this property, a function \mathcal{R} is devised that can be used to compare two algorithms. The common points between Ω and ω_A are given by:

$$\Omega \cap \omega_A$$

Now, if the algorithm is ideal:

$$|\Omega \cap \omega_A| = |\omega_A|$$

But in general,

$$|\Omega \cap \omega_A| \leq |\omega_A|$$

and

$$\frac{|\Omega \cap \omega_A|}{|\omega_A|} \leq 1$$

Thus, we can define the function \mathcal{R}_A for algorithm A as shown in Equation 20.

$$\mathcal{R}_A = \frac{|\Omega \cap \omega_A|}{|\omega_A|} \quad (20)$$

For comparing two algorithms A and B , \mathcal{R}_A and \mathcal{R}_B are calculated. If $\mathcal{R}_A > \mathcal{R}_B$, algorithm A is better than B and vice versa. If $\mathcal{R}_A = \mathcal{R}_B$, no conclusions can be derived from this metric and other properties have to be considered.

The function \mathcal{R} can also be used to track the convergence behavior of the algorithm through the generations. For each generation (or generation interval) g of the algorithm, \mathcal{R}_g can be calculated. These values can be plotted against the generation number to observe the convergence.

One important assumption behind the application of metric \mathcal{R} is the availability of the true Pareto set Ω . When Ω is not available other metric has to be used, such as the \mathcal{C} function proposed by Zitzler et al. [103].

5.7.2 Implementation of Algorithms

The implementation of SPEA2 is exactly based on the ideas explained by Zitzler et al. [108] and as described previously in this chapters.

For NPGA implementation, general ideas described previously are followed except few changes in the way sharing is implemented. Here, the maximum and minimum value of each objective function in a population set (for each generation) is determined. From these values, the ranges for all the objectives for that particular generation are calculated. Now an n -dimensional hypercube is formed around the two candidate points. The measure of each edge of this hypercube is equal to

$range/(PopulationSize)$ in that particular dimension. Now, sharing is executed by counting the number of population points that are present within the hypercube around each candidate points. The candidate with minimum number of neighbors represents a sparse region on the Pareto surface and is selected for next generation. This arrangement for sharing eliminates the need for specifying σ_{sh} and also to measure the Euclidian distance to perform sharing. As the hypercube constructed for sharing has dimensions relative to the objective values, the need to scale the objectives is avoided. Another advantage of this approach is that the dimensions of hypercube constructed are dynamic in nature and change from generation to generation; it is more representative of the nature of current population. Before sharing, Pareto dominance tournament takes place and the size of comparison set is fixed at 15 for higher dominance pressure.

One of the main difference between two algorithms is that SPEA2 uses archiving and NPGA does not. Even though the function \mathcal{R} is a normalized metric that is independent of the cardinality of ω , it is decided to keep the size of ω generated by both algorithms the same. Hence, in the interest of fair comparison, the population size for NPGA is same as the archive size of SPEA2. The parameters adopted for both algorithms are as follows:

Mutation Rate	: 0.05
Crossover Rate	: 0.8
Number of Generations	: 100
Population Size (NPGA)	: 300
Population Size (SPEA2)	: 100
Archive Size (SPEA2)	: 300

5.7.3 Simulation Results

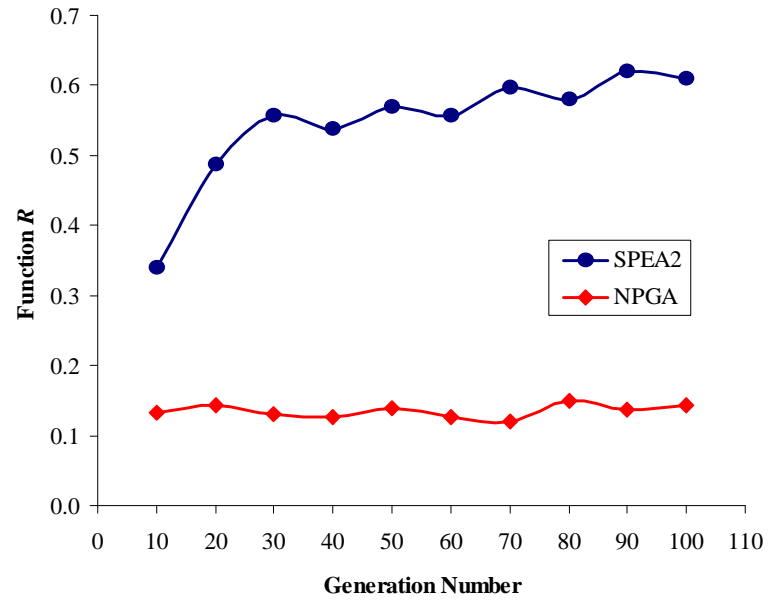
The function \mathcal{R} is evaluated for both the algorithms at the end of 100 generations. For NPGA $\mathcal{R}_{NPGA} = 0.143$, and for SPEA2 $\mathcal{R}_{SPEA2} = 0.61$. Thus, $\mathcal{R}_{SPEA2} > \mathcal{R}_{NPGA}$ and hence SPEA2 gives the better results.

To check the convergence of both algorithms, \mathcal{R}_{NPGA} and \mathcal{R}_{SPEA2} are calculated at the interval of 10 generations. The results are listed in Table 12. The first column under each algorithm lists the number of points that are common to Ω and ω , while the second column shows the value of function \mathcal{R} at the corresponding generation. The results are visualized in Figure 34. It can be observed from this figure that NPGA is not able to find Pareto optimal points. Throughout the generations, it behaves more like a random algorithm rather than consistently finding better solutions as the algorithm progresses. Moreover, the \mathcal{R} value is considerably lower for NPGA than that for SPEA2. On the other hand, the \mathcal{R} value for SPEA2 rapidly increases for initial 30 generations and then gradually converges to a value of around 0.6. This stark contrast in the performance of the two algorithms can be attributed to the fact that SPEA2 implements elitism via archiving while NPGA does not. Because of this, good solutions discovered by SPEA2 over the generations are not lost and are retained in the archive. Thus for SPEA2, genetics is not the only means of passing on information.

It is also interesting to look at the relative position of solutions offered by each algorithm in the objective space. For simplicity of visualization, two out of five dimensions are selected. These are objective 1 and 2 selected from Table 10 on the basis of the dominance based dimensionality reduction procedure described in Chapter 4. Non-dominated solutions in these two dimensions from each algorithm are plotted in Figure 35. Solutions from the true Pareto set are also superimposed on the plot. There are 34 non-dominated solutions for these two dimensions from the superset of 4,005 true Pareto solutions. There are 21 non-dominated solutions

Table 12: Convergence of NPGA and SPEA II

Generation	NPGA		SPEA2	
Number	$ \Omega \cap \omega $	\mathcal{R}	$ \Omega \cap \omega $	\mathcal{R}
10	40	0.133	102	0.340
20	43	0.143	146	0.487
30	39	0.130	167	0.557
40	38	0.127	161	0.537
50	42	0.140	171	0.570
60	38	0.127	167	0.557
70	36	0.120	179	0.597
80	45	0.150	174	0.580
90	41	0.137	186	0.620
100	43	0.143	183	0.610

**Figure 34:** Convergence of SPEA II and NPGA

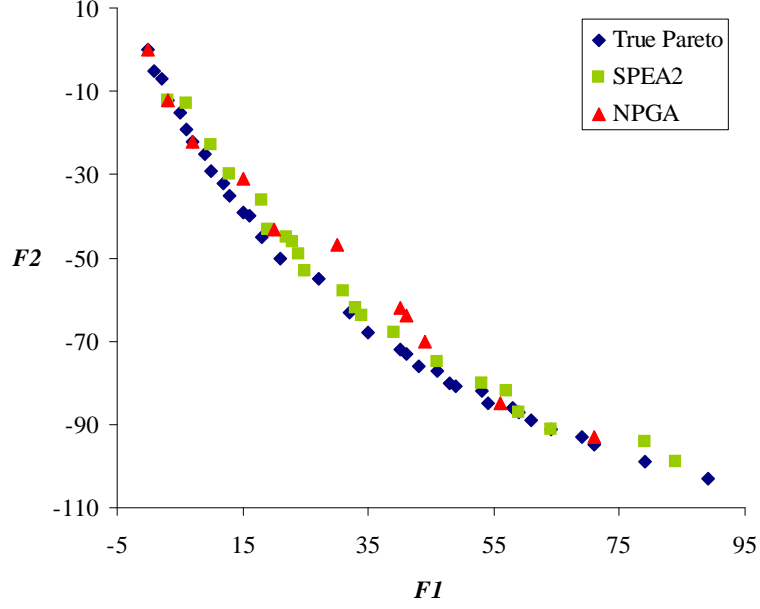


Figure 35: Comparing SPEA II and NPGA with True Pareto Solutions

in two dimensions from SPEA2 and 11 from NPGA results. As observed from the plot, SPEA2 solutions more accurately represent the actual Pareto front. The NPGA solutions are distant from the true frontier, especially in the central region. In terms of the distribution along the frontier, both algorithms provide evenly spaced points.

The runtime on a 2 GHz Pentium 4 machine for NPGA was around 54 seconds and for SPEA2 was about 184 seconds. This is because SPEA2 has more overheads in terms of calculating fitness based on dominance characteristics of each point in the population as well as in the archive. Moreover, for the knapsack problem, the time required for function evaluation is very less. For more complex function evaluations, the time advantage offered by NPGA would disappear as the total algorithm time would be governed by function evaluations rather than by the algorithmic operators.

These results have demonstrated that SPEA2 is a better choice for knapsack type problems. The following section describe further simulations carried out using SPEA2 to improve upon the previous results.

5.8 Efficacy of SPEA II

Having selected SPEA2 for Pareto optimization, it is important to check the parameter settings for which SPEA2 provides the best results. The most important parameters for this type of algorithms are the maximum number of generations, population size, and archive size. This section discusses the impact of varying these parameters for the benchmark problem. As in the previous section, the function \mathcal{R} is used as a measure for comparison. The results from SPEA2 with the best parameter settings are later compared with non-dominated results from a random search.

5.8.1 Effect of Changing Algorithmic Parameters

The following results are obtained by varying the parameters of SPEA2 algorithm. To check the impact of number of generations and population size, it was decided to fix the total number of function evaluations and the archive size. The function evaluations are fixed at 15,000 and the archive size at 500. For checking the impact of archive size, the total function evaluations are fixed and only the maximum archive size changed. Other parameters such as the crossover and mutation rates are fixed at the previously mentioned values.

5.8.1.1 Maximum Generation

For this simulation, the population size considered is 100 and the algorithm is continued through 150 generations. Thus $100 \times 150 = 15,000$ function evaluations. The archive at every 10th generation is retained and \mathcal{R} is calculated for that archive. The convergence for this simulation is illustrated in Figure 36. It is interesting to observe from the figure that the \mathcal{R} value increases steadily through generation 60. After that it fluctuates a little through generation 100 where it achieves the maximum value of 0.71. The time required for this simulation was about 9 minutes.

After generation 100, the \mathcal{R} value does not improve and settles at around 0.67. Thus, even though more new item combinations are evaluated after generation 100,

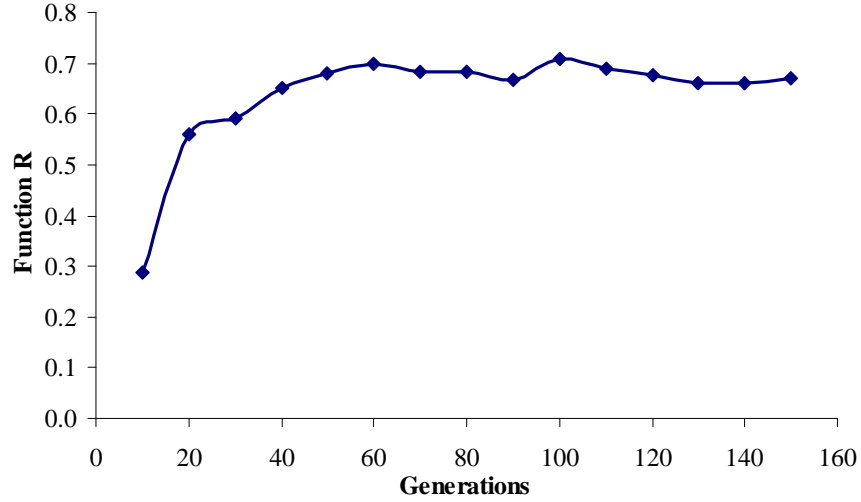


Figure 36: Convergence of SPEA II Through Maximum Generations

there is no improvement in the results. This observation is counterintuitive, especially given the elitism implemented using the archive. The main function of the archive is to retain the best solutions, yet there is a noticeable degradation in the results after generation 100. This phenomenon can be attributed to the limit on the archive size. Once this limit is reached during simulation, the algorithm selects new archive points based not only on non-domination but also based on niching. Thus, in the interest of even distribution of points along the frontier, some of the *true* non-dominated points are lost.

5.8.1.2 Population Size

This simulation is carried out with 100 generations and 150 population size. For the purpose of comparison, the convergence plot of this simulation is superimposed on the one for the previous simulation and illustrated in Figure 37. The \mathcal{R} value for this simulation at the end of generation 100 is 0.66. This is considerably lower than $\mathcal{R} = 0.71$ obtained at the end of 100 generations for the previous simulation with population size of 100. Moreover, the \mathcal{R} value plot for the current simulation after generation 30 is consistently below the one for the previous simulation.

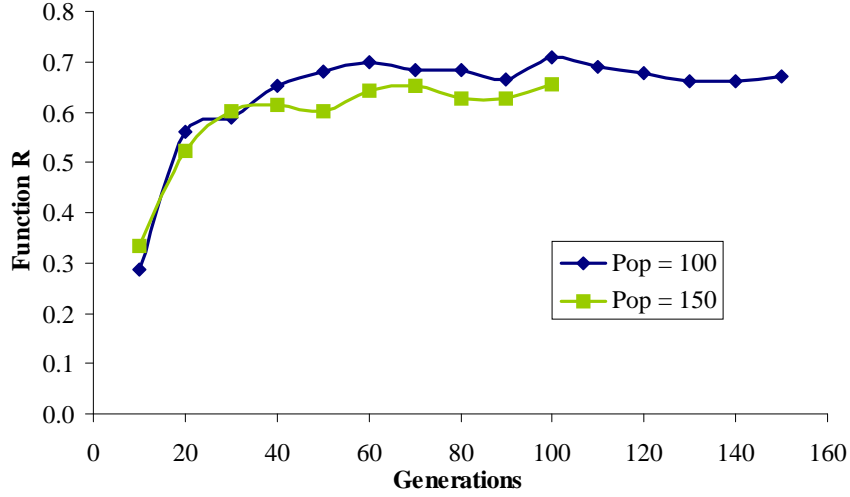


Figure 37: SPEA II Convergence for Different Population Size

The time required for this simulation was about 10.5 minutes, which is longer than the time required for simulating with a population of 100 for 150 generations. These results show that there is no advantage of increasing the population size independently.

5.8.1.3 Archive Size

The archive size is one of the most important parameter for a Pareto optimizing algorithm. To check the impact of this parameter on the results, three simulations are carried out with archive size of 300, 500, and 750. The population size and maximum generations remain fixed at 100 each. Thus, there are 10,000 function evaluations for each simulation. The convergence history of these simulations are illustrated in Figure 38. It is observed from this plot that as we increase the archive size, the quality of results improve. The main reason for this behavior is that with increasing archive size, the algorithm can retain more and more non-dominated solutions and does not need to prune the archive by performing niching operation. As a result more and more non-dominated solutions that are part of the *true* Pareto front are retained and hence higher \mathcal{R} values are observed for larger archive size.

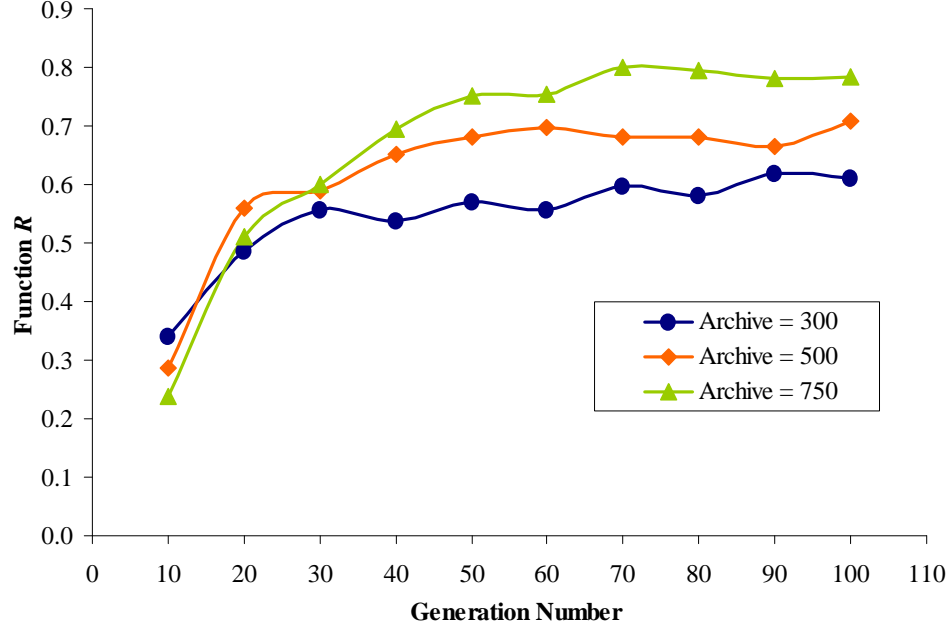


Figure 38: Impact of Archive Size on SPEA II Results

Thus, if limited number of function calls are allowed due to time limitations or other factors, the results can be improved by increasing the archive size. Moreover, results cannot be improved just by increasing the number of function evaluations; there has to be a corresponding increase in the archive size.

5.8.2 Comparing SPEA II with Random Search

When designing a Pareto optimization algorithm, it is important to compare its results with random search. For this purpose, an experiment is conducted by simulating SPEA2 algorithm for 100 generations with 100 population and 1200 archive size. As observed in the previous section, larger archive size helps retain good solutions without increasing the number of function evaluations. The results from this simulation are compared with results from a random search conducted with 10,000 function evaluations. The number of function evaluations is same for SPEA2 and random search in order to understand the efficacy of SPEA2. In case of SPEA2, 980 points from the final archive are part of the true Pareto front of 4,005 points. On the other hand,

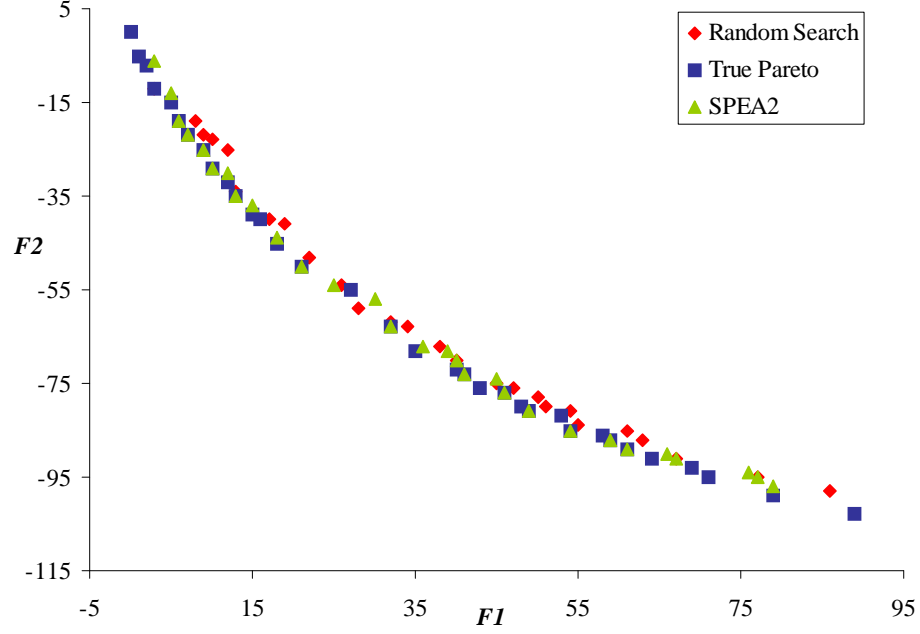


Figure 39: Comparing SPEA II with Random Search in 2 Dimensions

only 532 points from randomly selected 10,000 points are part of the true Pareto front.

To have a better idea of the difference in two approaches, a 2-dimensional case is considered with objective 1 and 2. In these dimensions, there are 29 non-dominated points from SPEA2 results and 23 from the random search. These points are plotted in Figure 39 along with the 34 non-dominated points representing the true Pareto front in two dimensions. When compared with the true Pareto front, there are 13 points from SPEA2 results that are part of the true frontier. While only one point from the random search results is part of the true Pareto front. It is also observed from the figure that SPEA2 results better represent the Pareto front. Thus, for a given number of function evaluations, SPEA2 results are much more accurate than the random search results.

5.9 Summary

It was observed in the previous chapter that *a posteriori* preference articulation framework is better suited for the multi-objective technology selection problem. One of the most challenging task in implementation of this framework is the search for a representative subset of the Pareto optimal solutions. Evolutionary algorithms are known to be very efficient for this task of Pareto optimization. The concepts behind this and various algorithms available for the task are studied in this chapter. It is hypothesized that SPEA2 is the most appropriate algorithm for searching the Pareto optimal points in a multi-dimensional combinatorial technology space. To check the plausibility of this hypothesis, experiments are conducted using a benchmark knapsack problem.

Initially, results from SPEA2 are compared with another promising algorithm called NPGA. A comparison metric is devised that provides a quantitative value to how well the results from an algorithm represent the true Pareto frontier. The results from SPEA2 are observed to be much better than the ones from NPGA. The convergence behavior of SPEA2 with different parameter setting is checked using the comparison metric. Archive size is observed to be the most important parameter and it has considerable impact on the accuracy of the result. In the end, the results from SPEA2 are compared with the results from a random search. It is observed that given a fixed number of function evaluations, the results from a random search are not very encouraging; on the other hand, the Pareto front from SPEA2 accurately represents the true Pareto frontier.

CHAPTER VI

PROBABILISTIC TECHNOLOGY SELECTION

The impact of a technology on the system is never deterministic. There are always uncertainties involved, even when the technologies under consideration are mature. These uncertainties demand due attention while selecting technologies to be included in a system. This chapter investigates one of the primary research questions posed earlier: *How to account for technological uncertainties while selecting technology combinations?* An attempt is made to address this question considering the implementation of *a posteriori* preference articulation framework for multi-objective decision making.

The chapter starts with an overview of probabilistic design as related to technology selection. Description of the technological uncertainties under consideration is provided. Technique to represent these uncertainties is explained. Techniques for probabilistic analysis are reviewed and the most promising is selected. Based on this technique and previously discussed *a posteriori* preference articulation framework, a novel approach for probabilistic technology selection is proposed. The soundness of this approach is verified using a benchmark knapsack problem.

6.1 Probabilistic Design

The field of engineering design that deals with uncertainties in the parameters and their impact on the system responses is known uncertainty-based design. As described in the NASA white paper [110] on the topic, the uncertainty-based design is used to describe design problems that have non-deterministic problem formulation. In non-deterministic problem formulations, some essential components of the problem are treated as non-deterministic; that is, some form of variability is associated with these

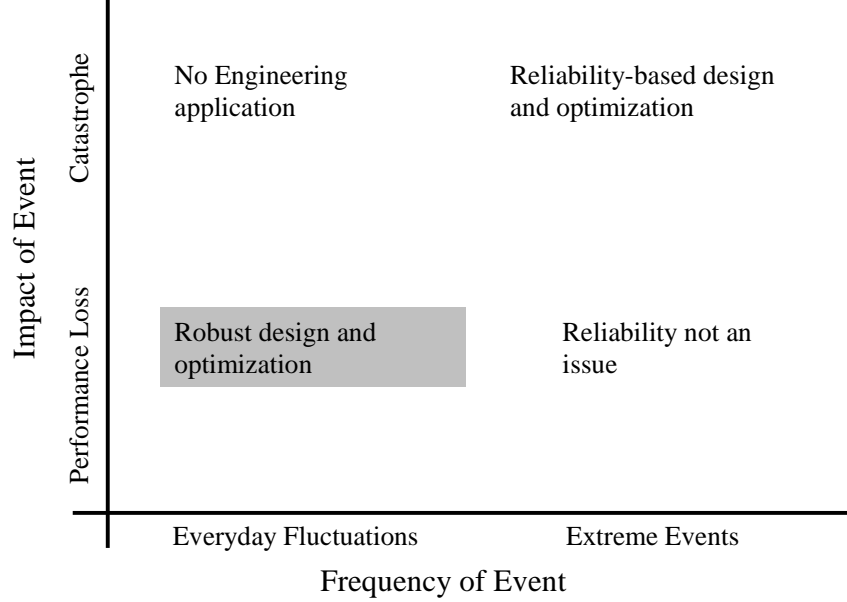


Figure 40: Facets of Uncertainty-Based Design

components. They can be, for example, noise variables such as tolerances in manufacturing processes, uncertain input parameters, simulation and experimental errors, etc. The problem addressed in this thesis also has a non-deterministic formulation because of the uncertainties associated with technology impacts on the system.

There are two main facets of uncertainty-based design – Robust design and Reliability-based design. The difference between these two problems is illustrated in Figure 40 (adapted from Huyse [111]). Reliability-based design deals with extremely rare events with catastrophic impact on the system. The design concepts in this category are originated from the field of structural engineering where it is required to design a component/system that has probability of failure less than some accepted (invariably small) value. On the other hand, Robust design deals with problems where insensitivity to the variations in uncertain parameters is desired. Thus, reliability-based design is more concerned with the extremes of the distribution while in case of robust design, the designer is more interested in the central part of the distribution, as illustrated in Figure 41 (adapted from Zang et al. [110]).

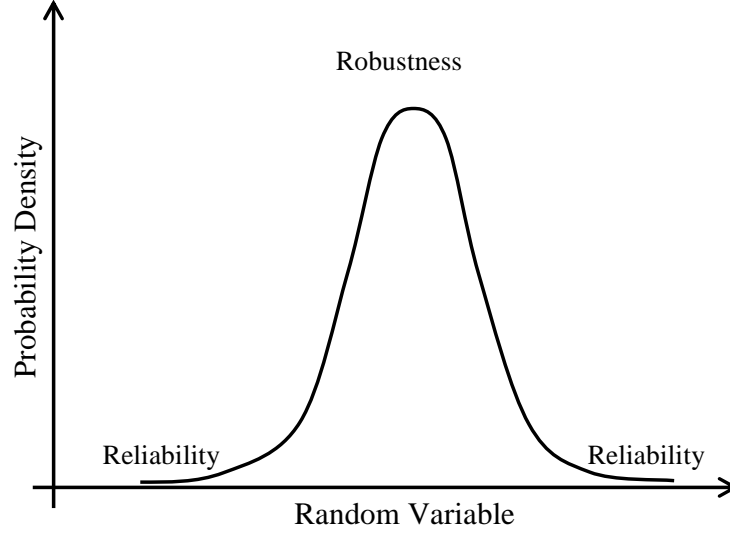


Figure 41: Robust Design and Reliability-Based Design

For the technology selection problem, it is desired that the resultant system designed with selected technology combination is insensitive to the variations in the impacts of technologies included. By this interpretation, it is a robust design problem. At the same time, the system needs to satisfy certain performance and economic targets; and the designers are interested in knowing the probability of achieving those targets. If these targets are not achievable with a high level of confidence, the design is considered infeasible or inviable. In this perspective, the problem is more than just of robust design. To capture this characteristic, a more generic term of Probabilistic Design is used to describe the technology selection problem. There are three main ingredients involved in probabilistic design problems:

- Quantification of parameter or input uncertainties.
- Quantification of system level or output uncertainties using various probabilistic analysis techniques.
- Design optimization over system level uncertainties.

These three steps as applicable for probabilistic technology selection are described in the following sections.

6.2 *Technological Uncertainties*

The impact of technologies on the system is quantified in terms of changes in key parameters known as technology metrics or *k*-factors [13]. For this, the most important *k*-factors for a given system are identified and functionally related to the overall system performance through system models or surrogate models. These *k*-factors are combined to form a Technology Impact Matrix (TIM) that map technologies to the *k*-factor space, which in turn map the system performance through system models. This mapping is illustrated in Figure 42. Thus in essence, technologies are mapped to system performance in two steps. Now, all the technologies involved are at different Technology Readiness Level (TRL). The ones with lower TRLs generally have more variability associated with them as they are not yet fully understood. While, the ones with higher TRLs also have some variability associated with them; because, even though they are more mature, their impact on the system may not be fully understood. Thus, the impact of a technology on the *k*-factors is uncertain and this uncertainty is propagated through the system responses. These uncertainties have to be defined and adequately represented.

6.2.1 *Epistemic Uncertainty*

Uncertainties are divided into two main categories, first being *epistemic uncertainty* and the other being *aleatory uncertainty*. Aleatory uncertainty arises because the system under study may naturally behave in several different ways, i.e., uncertainty due to random processes. Epistemic uncertainty, on the other hand, arises due to insufficient knowledge [112]. Epistemic uncertainty is also called subjective, state of knowledge, or reducible uncertainty and aleatory uncertainty is known as stochastic, inherent or irreducible [113]. The technological uncertainties arise mainly because the impact of technologies captured by the *k*-factors is generally based on the subjective assessments of respective technology experts. This introduces a margin of error in

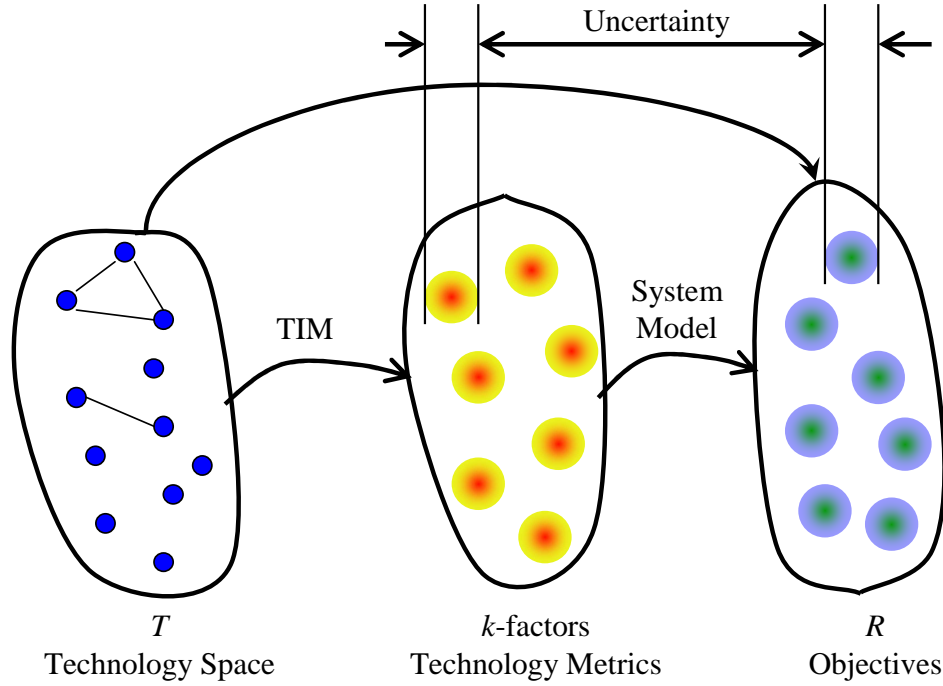


Figure 42: Mapping Technologies to the System

the technology metric values which is propagated through the system performance.

There has been a resurgence in the study of epistemic uncertainty in the recent years as evident by the Epistemic Uncertainty Project sponsored by the Sandia National Laboratories. A set of challenge problems were designed under this project to study various approaches for accounting epistemic uncertainty in system modeling [114]. As compared to aleatory uncertainty which is generally described using probability distributions based on experimental or statistical data, mathematical representation of epistemic uncertainty is a challenge. Apart from the traditional probability theory, other approaches, grouped together into Generalized Information Theory (GIT) [115] are available to address this issue. Helton et al. [116] explore various approaches of GIT such as evidence theory, possibility theory, and evidence analysis in addition to probability theory for uncertainty representation. Possibility theory is used by Chae [117] for sizing a rotorcraft system in the absence of complete information. O'Hagan and Oakley [118] argue that the best approach for representing

and quantifying all forms of uncertainty is through the traditional probability theory. This argument is strengthened by the fact that alternative theories face considerable conceptual challenge for propagating the parameter uncertainty through the system model, while this is mathematically well formulated for probability theory. According to them, the only thing that needs to be further studied for applying probability theory for epistemic uncertainty is the practical and sufficiently accurate elicitation of expert knowledge. This is further discussed in the following subsection.

6.2.2 Uncertainty Representation

As mentioned before, uncertainty is associated with each element of the TIM and this section describes technique to represent that uncertainty. Uncertainty representation starts with the data gathered from technology experts. Batson and Love [119] have formulated a method of encoding subjective responses from technology experts into beta distributions. This method has been successfully adopted by Kirby et al. [21] for the TMAAT process.

6.2.2.1 Beta Distribution

Beta distribution is defined over the interval $[0, 1]$ and its most common application is in modeling proportions [120]. The Probability Density Function (PDF) of beta distribution is given by Equation 21.

$$f(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)} \quad \text{for } 0 \leq x \leq 1 ; \quad \alpha, \beta > 0 \quad (21)$$

Here, $B(\alpha, \beta)$ is called the beta function with shape parameters α and β .

$$B(\alpha, \beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1} dt$$

The general beta distribution is defined for the closed interval $[a, b]$ and its PDF given by Equation 22.

$$f(x; \alpha, \beta) = \frac{(x-a)^{\alpha-1}(b-x)^{\beta-1}}{B(\alpha, \beta)(b-a)^{\alpha+\beta-1}} \quad \text{for } a \leq x \leq b ; \quad \alpha, \beta > 0 \quad (22)$$

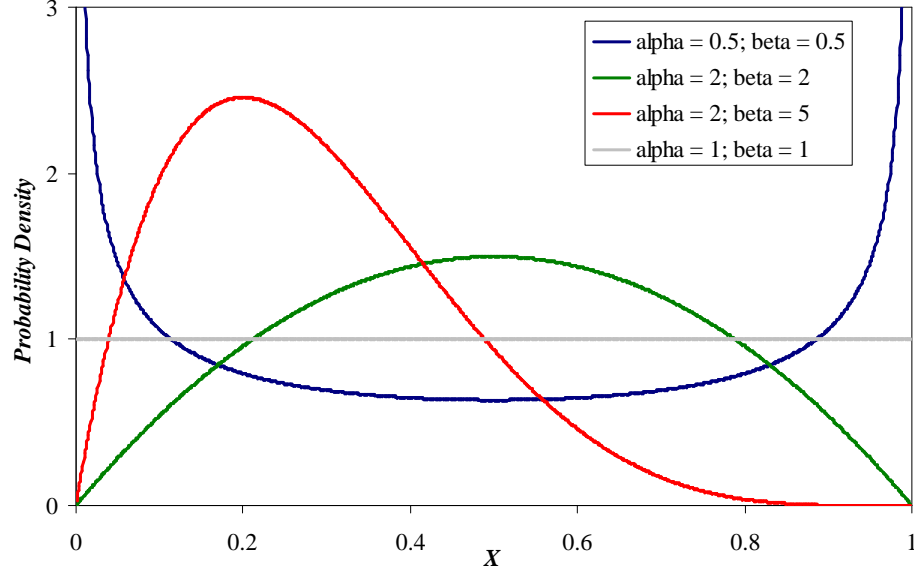


Figure 43: PDF of Beta Distribution with Different Parameter Values

When $a = 0$ and $b = 1$, the beta distribution is known as the *standard beta distribution*. By using location a and scale parameters $b - a$, any general beta distribution can be expressed in terms of standard beta distribution.

Depending on α and β values the distribution can take a variety of forms. This property is illustrated in Figure 43. Because of this property, a single probability formulation can be used to describe various shapes of probability distributions associated with the technology impacts. Moreover, even though originally defined for the interval $[0, 1]$, it can be extended to any finite interval using the generalized form. This flexibility that beta distribution provides makes it the preferred distribution for quantifying subjective probabilities [121].

6.2.2.2 Technological Uncertainty With Beta Distribution

Kirby et al. [21] have proposed a Technology Audit scheme to elicit information about technological uncertainties from respective technology experts. The experts provide the *maximum* (max), *minimum* (min) and *most likely* (ml) values for the technology metrics (k -factors). These values can be the actual metric values or proportional

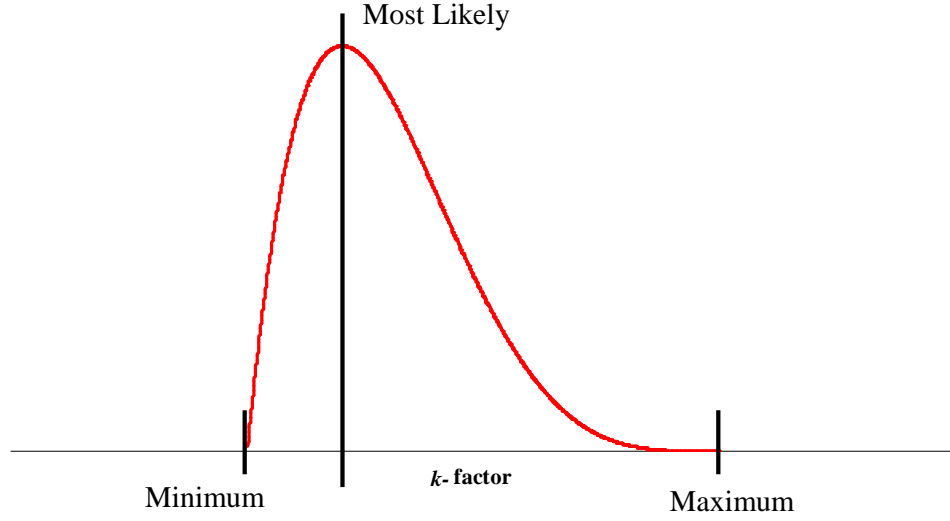


Figure 44: Beta Distribution from Elicited Values

values with respect to a fixed technology baseline. These three values are used to define a beta distribution for each technology metric. A notional representation for this process is illustrated in Figure 44. The authors suggest using an iterative process, similar to the Delphi method [122], for Technology Audits to ensure that the distribution created is a realistic representation of the expert opinion.

The Technology Audit is conducted every year and the distributions are updated till the technology matures and the system is developed. Thus, these distributions change through the years as technology matures and is better understood. This change can be tracked to forecast the future progress of technology. Even though this is an important phenomenon that the system designers have to be aware of; for the purpose of this research, the distribution is assumed to be fixed. That is, one distribution is used for to account for the uncertainty in the impact of a technology on one k -factor.

Project Evaluation and Review Technique (PERT) [123] approximations help calculate the mean and variance of the elicited values as given by Equations 23 and 24

respectively.

$$mean = \frac{min + 4ml + max}{6} \quad (23)$$

$$variance = \sigma^2 = \left(\frac{max - min}{6} \right)^2 \quad (24)$$

The first two moments, expected value and variance, of a general beta distribution are defined by Equation 25 and 26 respectively.

$$E(x) = a + \frac{\alpha}{\alpha + \beta} \times (b - a) \quad (25)$$

$$Var(x) = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)} \times (b - a)^2 \quad (26)$$

Equating the mean and variance from the PERT approximations with Equation 25 and 26 respectively, one can obtain the shape parameters of the beta distribution. These are given by Equation 27 and 28.

$$\alpha = \left(\frac{mean - a}{b - a} \right) \left(\frac{(mean - a)(b - mean)}{\sigma^2} - 1 \right) \quad (27)$$

$$\beta = \left(\frac{b - mean}{mean - a} \right) \times \alpha \quad (28)$$

This technique, based on the PERT approximations and the moments of beta distribution, provides a relatively straightforward approach of extracting probability distribution out of subjective estimates on technology impact. Other approaches for estimating mean and variance can also be used, as suggested by Perry[124] and Keefer[125], while using the overarching framework of equating them to the moments of preferred distribution.

6.3 Probabilistic Analysis

Once the uncertainties are defined for different product and program inputs, system level uncertainties have to be quantified via probabilistic analysis. There are various methods and techniques available for this task, many of them coming from the field of structural reliability. Robinson [126] presents a comprehensive survey of probabilistic

methods used for engineering design. This section describes some of the reliability based techniques that can be used for system level probabilistic analysis. A more preferred approach using Monte Carlo Simulations (MCS) is also explained. Finally, probabilistic analysis frameworks using the analysis techniques are discussed.

6.3.1 Convolution

If X and Y are independent random variables with distribution α and β , then the distribution of $Z = X + Y$ is given by the *convolution* of α and β , and is denoted by $\alpha * \beta$. In terms of characteristic function ϕ , the convolution is expressed as:

$$\phi_{\alpha*\beta}(t) = \phi_{\alpha}(t)\phi_{\beta}(t)$$

where,

$$\phi_{\omega}(t) = \int e^{itx} d\omega$$

Moreover, if $f(x)$ is the PDF for random variable X , the characteristic function is defined as:

$$\phi(t) = \int_{-\infty}^{\infty} e^{itx} f(x) dx$$

which is the Fourier transform of the PDF. The PDF of Z is computed as:

$$f(z) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{itz} \phi_{\alpha*\beta}(t) dt$$

from which the CDF of Z can be computed.

This method provides a direct way of calculating the PDF for the sum (or linear combination) of any number of independent random variables. One way of computing $\phi_{\alpha*\beta}$ and $f(z)$ is by using numerical integration, but this may be computationally expensive. A more efficient approach is the use of discrete fast Fourier transform (FFT) as suggested by Wu [127]. The basic procedure involves discretizing the PDFs of independent variables. The discretized characteristic function of these variables is obtained by applying FFT on these PDFs. The product of these characteristic

functions results in the discretized characteristic function of Z . Finally the discretized PDF of Z is evaluated by the inverse FFT (IFFT) of its characteristic function. The method can be adapted for nonlinear systems with dependent random variables. For example, Sakamoto[128] and Penmetsa[129] have applied the convolution theorem with FFT to solve structural reliability problems with implicit limit state functions.

6.3.1.1 *Advantages and Shortcomings*

The method is theoretically sound and errors are introduced only while approximating the state function when it is implicit or highly nonlinear and while discretizing the PDFs. It can be applied for any number of variables with any type of distributions. This method requires less function evaluations compared to methods such as MCS, but computational time may increase significantly as the number of variables increase; this is due to the discrete nature of FFT. Moreover, it is required to linearize the response function.

6.3.2 **Mean Value Methods**

Mean value methods are probabilistic analysis methods based on the concept of limit state function. Lets say $Z(X)$ (Z -function) is a response or performance function of X_i random variables. Now a g -function is a limit state function defined as:

$$g(X) = Z(X) - z_0 = 0$$

where z_0 is a specific value of Z , any value below this is not desirable. Thus the g -function defines the boundary $g(X) = 0$ that divides the failure ($g \leq 0$) and safe ($g > 0$) region of the design space. By varying z_0 a series of limit states can be formed that can be used to create a complete CDF for Z function.

The basic mean value method assumes that the Z -function is smooth and Taylor's

series expansion given by Equation 29 exists at the mean values of variables.

$$\begin{aligned} Z(X) &= a_0 + \sum_{i=1}^n a_i X_i + H(X) \\ &= Z_{MV}(X) + H(X) \end{aligned} \tag{29}$$

Approximate mean and standard deviation of Z are computed by using only the first-order terms of the expansion Z_{MV} and higher order terms $H(X)$ are neglected. This technique is called mean value first order (MVFO) or simply MV. For problems involving highly nonlinear or implicit Z functions advance mean value (AMV) provides a better solution. AMV method uses a simple correction procedure to compensate for the truncation errors present in the MV method by replacing $H(X)$ by a simpler function $H(Z_{MV})$. This is accomplished by using the concept of Most Probable Point (MPP). It is the design point defined in independent and normalized parameter space; a detailed explanation of this concept is provided in the FPI manual [127]. The number of function evaluations for AMV method to obtain a probability distribution is $n + m + 1$ where n is the number of random variables and m is the number of probability levels desired. Wu et al. [130] provide an in depth description of AMV method along with some numerical examples pertaining to structural reliability.

6.3.2.1 *Advantages and Shortcomings*

The mean based methods are computationally much more efficient than MCS or Convolution based methods. But, if a function is highly nonlinear, the number of points required to define the CDF may be high. AMV faces some limitations when the input parameters are highly correlated. Moreover, as Fox and Reh [131] have warned, mean value based methods cannot be blindly used without verifying their accuracy for a specific problem.

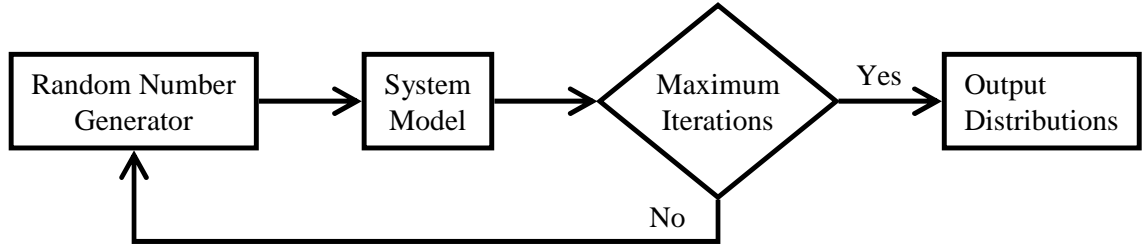


Figure 45: Monte Carlo Simulation

6.3.3 Monte Carlo Simulation (MCS)

Monte Carlo Simulation is a class of methods used to simulate stochastic processes in science, engineering, business, etc. and to numerically solve mathematical problems. MCS iteratively generates a set of random numbers from the Probability Density Functions (PDFs) of independent input parameters and computes corresponding system responses. These response values are used to construct output Cumulative Distribution Functions (CDFs) and PDFs. Dienemann [132] provides one of the earliest application of MCS in system design for estimating cost uncertainty. An outline of the method is illustrated in Figure 45.

6.3.3.1 Number of Samples Required

One of the main questions with MCS is: *How many samples are required for creating a sufficiently accurate CDF of the response function?* Bandte [133] provides a comprehensive answer to this question which is paraphrased here.

MCS do not provide an exact continuous distribution of a response function but rather, simulates this in the form of a discrete binomial distribution. The binomial distribution is obtained by a sequence of Bernoulli trials. The outcome of a Bernoulli trial is either 0 or 1. Processes with only two possible outcomes are represented by Bernoulli random variable, for example, a coin toss. Thus, n MCS runs are in effect n Bernoulli trials with n input samples. Each input sample i results in a certain

response value R_i . MCS tracks the number of trials that result in response values less than a specific value r . Let i^{th} Bernoulli trial is represented by x_i , and if the result of this trial is $R_i < r$, then $x_i = 1$, and if its not then $x_i = 0$. The true probability of $x_i = 1$ is denoted by p , i.e. $p = P(R < r)$.

The random variable $X = \sum_{i=1}^n x_i$ is said to have a binomial distribution with parameters n and p . This random variable takes the values from 0 through n and records the number of trials with $x = 1$. The mean and variance for X is given by $\mu = np$ and $\sigma^2 = np(1 - p)$ respectively.

For reasonably large n and p not too close to 0 and 1, normal distribution provides a good approximation for the binomial distribution [120]. Thus, $N(np, np(1 - p))$ represents the binomial distribution with corresponding mean and variance. As a general rule, this approximation is reasonable when $np \geq 5$ and $n(1 - p) \geq 5$. [120]

The normal random variable takes a value within two standard deviations of its mean with a 95% probability. Thus, for $Z \sim N(0, 1)$ (following a *standard* normal distribution),

$$P(-2\sigma \leq Z \leq 2\sigma) = 0.95$$

Now, X does not follow the standard normal distribution, hence it has to be transformed by subtracting the mean and dividing the value by standard deviation. Thus,

$$P\left(\frac{-2\sigma - \mu}{\sigma} \leq \frac{X - \mu}{\sigma} \leq \frac{2\sigma - \mu}{\sigma}\right) = 0.95.$$

Substituting for μ and σ , and dividing by n ,

$$P\left(-2\sqrt{\frac{p(1-p)}{n}} \leq \frac{X}{n} - p \leq 2\sqrt{\frac{p(1-p)}{n}}\right) = 0.95. \quad (30)$$

Now, to have an accurate representation of the response distribution, the sampled probability value X/n needs to be close to the real probability p . The Equation 42 defines the error ε associated with this approximation.

$$\varepsilon = \frac{\frac{X}{n} - p}{p} \quad (31)$$

From Equation 42 and 30, for a 95% confidence the maximum error is given by:

$$\varepsilon = 2\sqrt{\frac{(1-p)}{np}}$$

Solving the above equation for n ,

$$n = \frac{4}{\varepsilon^2} \frac{1-p}{p} \quad (32)$$

The required sample size for MCS for desired error and p values can be obtained from Equation 32. The sample size values as obtained from this equation are plotted against p values and illustrated in Figure 46. Each curve in the figure represents different error values. The sample size is depicted on a logarithmic scale in the figure. It can be observed from the figure that when very low probabilities ($p < 0.05$) are desired, the sample size required increases considerably and is prohibitive for most system analysis cases. On the other hand, when considering higher probabilities and higher error values, the sample size is very less and the sampling cannot be considered statistically significant [133].

6.3.3.2 Advantages and Shortcomings

The main advantage of this method is that it provides asymptotically exact solution as the number of iterations approach infinity. The disadvantage being that the computational cost may be too high, especially for complex response functions, to obtain very accurate results. On the other hand, if the accuracy of very small probability values is not required, this method can be very efficient. Moreover, it does not require the extra steps to transform the response functions as is required in other methods.

6.3.4 Probabilistic Analysis for Complex Functions

Based on the above analysis techniques, there are three basic probabilistic formulations as proposed by Fox [134] that can be used with complex analysis tools. These are illustrated in Figure 47 (adapted from [135]). The first formulation in the figure

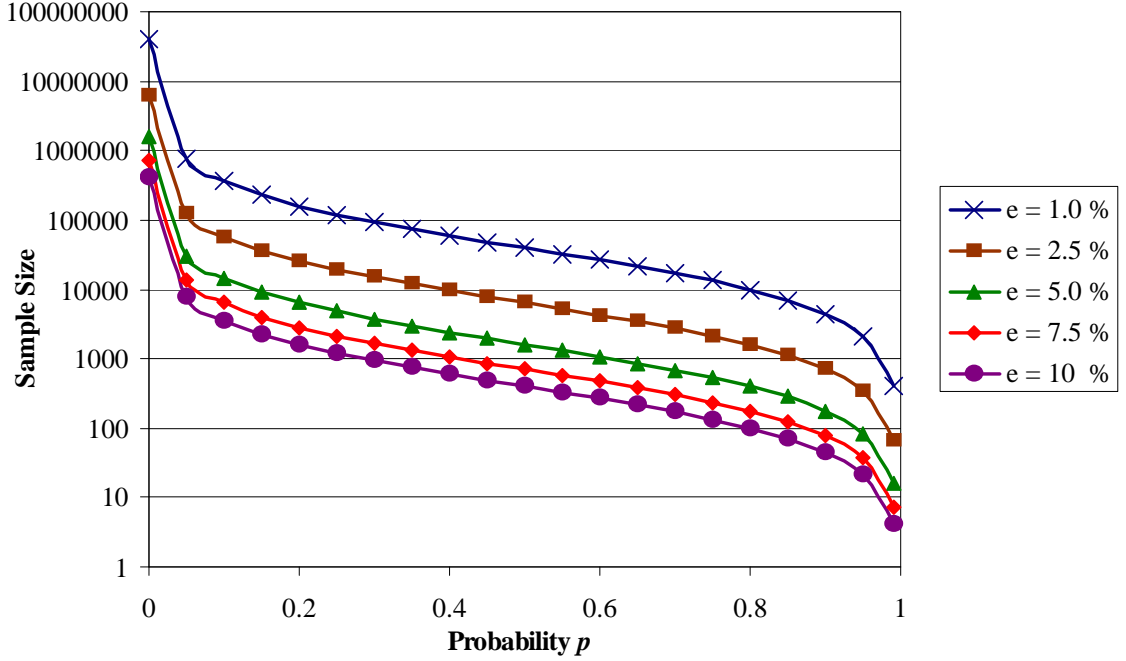


Figure 46: Sample Size Requirement for Monte Carlo Simulation

directly links the most accurate but computationally intensive probabilistic analysis technique, such as MCS, with the traditional system analysis tools used for technology evaluation. Given sufficient number of MCS runs, this method is the most accurate of all. But, it is also computationally very intensive and not preferred for the current problem.

The third formulation from Figure 47 also uses the exact system analysis tools; but in place of MCS, it uses a more efficient but approximate probabilistic analysis tool such as AMV method. This approximation of the response distribution is based on the notion that not all probability levels need to be identified in order to create a CDF. This formulation is extensively used in the field of structural reliability analysis as it is very efficient in analyzing the extremes of a response distribution [130, 136].

In the second formulation from Figure 47, the exact probabilistic analysis technique of MCS is used with approximations of the system models. These approximations are known as surrogate models and can be obtained using the Response Surface

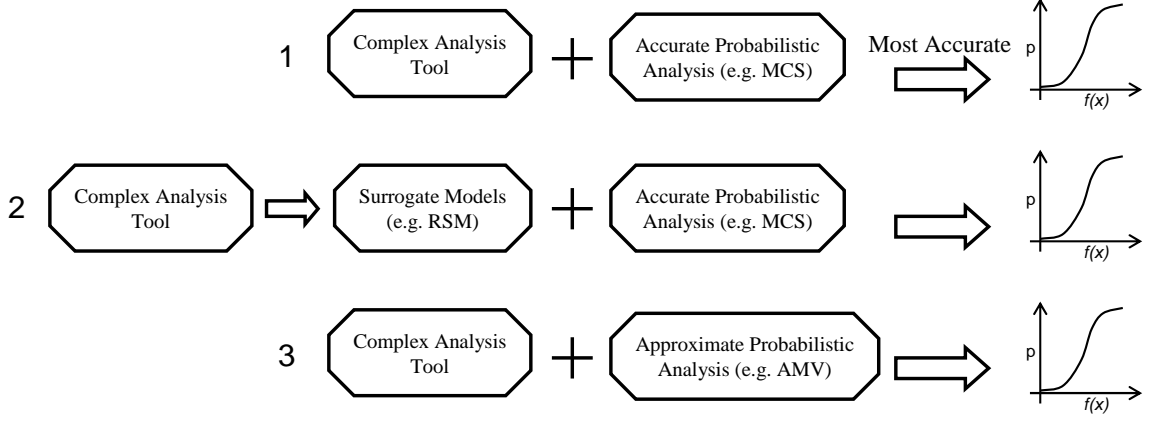


Figure 47: Probabilistic Analysis Methods

Methods [151] among other techniques. The surrogate models significantly reduce the computational time. Depending on the accuracy of the surrogate models, this formulation can be computationally efficient and yet provide very accurate CDFs. This is a more common approach used for robust design at the conceptual and preliminary design phases. For example, Mavris et al. [137] have developed Robust Design Simulation (RDS) using this approach for probabilistic design of an aircraft.

6.3.4.1 Probabilistic Analysis Framework for Technology Selection

The technology selection problem is computationally expensive to solve, primary reason being the exponential increase in the number of combinations with increasing number of available technologies. Thus, it was decided early on in this research that surrogate models created from the physics based aircraft analysis codes would be used to evaluate technologies. Given this availability of surrogate models, the second formulation from Figure 47 is preferred for probabilistic analysis of technology combinations. Moreover, as observed from Figure 46, a small sample size of less than 1000 points can be used to obtain a relatively accurate response CDF when the p -values of interest are more than 50%.

6.4 Probabilistic Optimization

As the uncertainty-based design domain has two main facets, the field of probabilistic optimization is also catered towards these two facets – *optimization for reliability* and *robust optimization*. The basic assumption behind optimization for reliability is that the design space is divided in two regions – success and failure. The goal of optimization is to find a design that is far away from the failure region, and the probability of failure is extremely small [110]. Mathematical optimizers are widely employed for the task as demonstrated by Eldred et al. [136]. On the other hand, the aim of robust optimization is to find a design that is insensitive to the parameter variations. In other words, the optimization process tries to find a design with narrow response PDFs. Generally, this is achieved by optimization routines trying to optimize mean and variance of the response at the same time. An interesting approach towards this is presented by Kumar et al. [138] using a 2-dimensional Pareto front; one dimension for mean and other for variance of the response.

For the purpose of probabilistic technology selection, there are some fundamental criteria based on which the designers and DMs would like to make decisions. These are stated as follows:

- A design that is insensitive to the variabilities in technology impacts. That is, a technology combination that results in the most narrow response PDFs.
- Knowing the probability of success with which a design with a certain technology combination meets the performance and economic targets.
- Selecting technology combination based on the level of confidence, say 90%. That is, response values corresponding to 90% probability level on the Cumulative Distribution Function (CDF) are used to compare different technology combinations and finally select one.

There are a couple of techniques that can satisfy some of the above requirements and be implemented within the *a posteriori* preference articulation framework. The first one is by Kumar et al. [138] noted previously. For the implementation of this technique, the already large dimensional problem space would double. This is because for each objective being considered, there will have to be two dimensions, one for the mean and the other for the variance. This will significantly increase the dimensionality and will have adverse consequences for the search algorithm. The other technique is of post-optimality probabilistic analysis as implemented by Adumitroaie et al. [27] and discussed in Chapter 2. For this, some of the promising technology combinations can be selected from the Pareto front and probability analysis carried out on them. In this case though, the initial selection of combinations is based on deterministic evaluation. Thus, all the available combinations in the design space are not compared probabilistically. To address these limitations of the existing approaches, a novel probabilistic technology selection framework is proposed.

6.5 Proposed Probabilistic Technology Selection Approach

The basic question now is: *How to address, in a comprehensive manner, technological uncertainties within the MODM framework of a posteriori preference articulation?* By the term *comprehensive manner*, it is expected that all technology combinations are compared probabilistically in the objective space *for* creating a Pareto optimal subset. This is different from probabilistically comparing the solutions *after* creating a Pareto optimal subset. It can be safely assumed that with the former method, the Pareto front would be different and more accurately represent the probabilistic results, as compared to the later method. In fact, because of the uncertainties in the technology impacts, each response of a technology combination behaves as a random number. Thus, the dominance characteristics of points change with probabilistic results as compared to the deterministic results, hence the Pareto front itself changes.

This is notionally illustrated in Figure 48. For this notional example, Pareto optimization with deterministic values will result in only one point, point a . Point b is dominated by point a as shown in Figure 48(a) and will not be part of the solution. Now, let's consider that both points are evaluated probabilistically and 80% probability level (p -level) is considered the selection criteria. The notional CDFs for this are shown in Figure 48(b). Because of this probabilistic analysis, the response values change for both points and now both are part of the Pareto set and hence part of the solution. If post-optimality probabilistic analysis was implemented, only point a would be the candidate and point b would not have been considered.

6.5.1 Joint and Marginal Probability Distributions

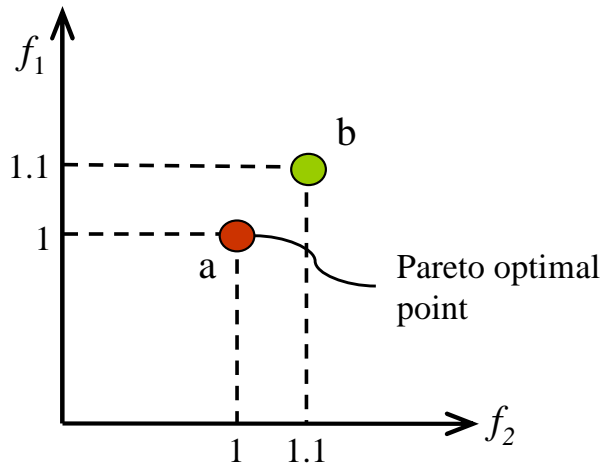
In the above example, two CDFs (one for each response) for each point are used to fix the p -level value for that point. This leads to questions regarding the nature of distribution on a point in the multi-dimensional objective space. *Is this the right way to represent uncertainties in the responses?* As the technology decisions are based on multiple objectives rather than just one, can the uncertainties in each objective be considered individually?

Given the uncertainties in technology impacts and their propagation through the responses, each point representing a technology combination is going to be *jointly distributed* in the objective space.

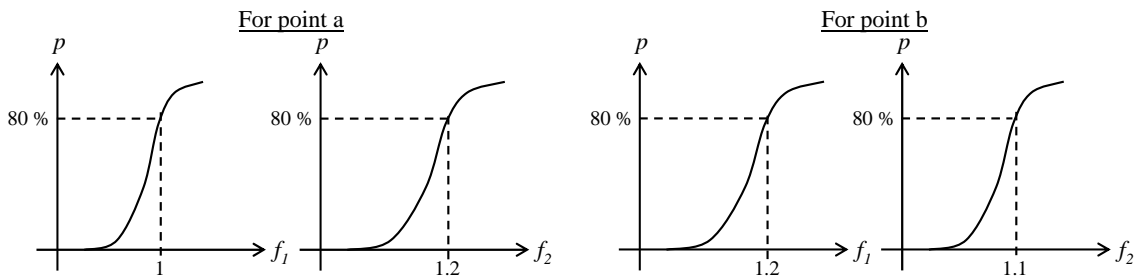
Definition: The **Joint Probability Distribution** for two continuous random variables X and Y is specified by the joint probability density function $f(x, y)$ and the joint cumulative distribution function is given by:

$$F(x, y) = P(X \leq x, Y \leq y)$$

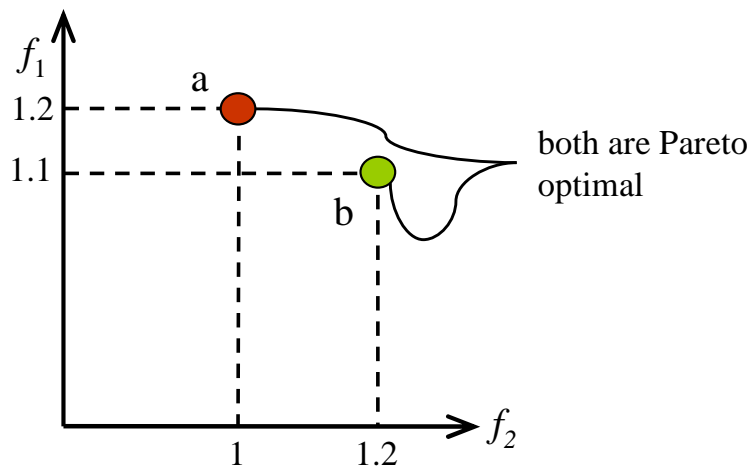
Now, even though two variables are jointly distributed, it is appropriate to focus on only one variable at a time. The probability distribution of this random variable is called the *marginal distribution*.



(a) With Deterministic Evaluation



(b) Probabilistic Analysis of Two Points



(c) With Probabilistic Evaluation

Figure 48: Change in Dominance Structure with Probabilistic Results

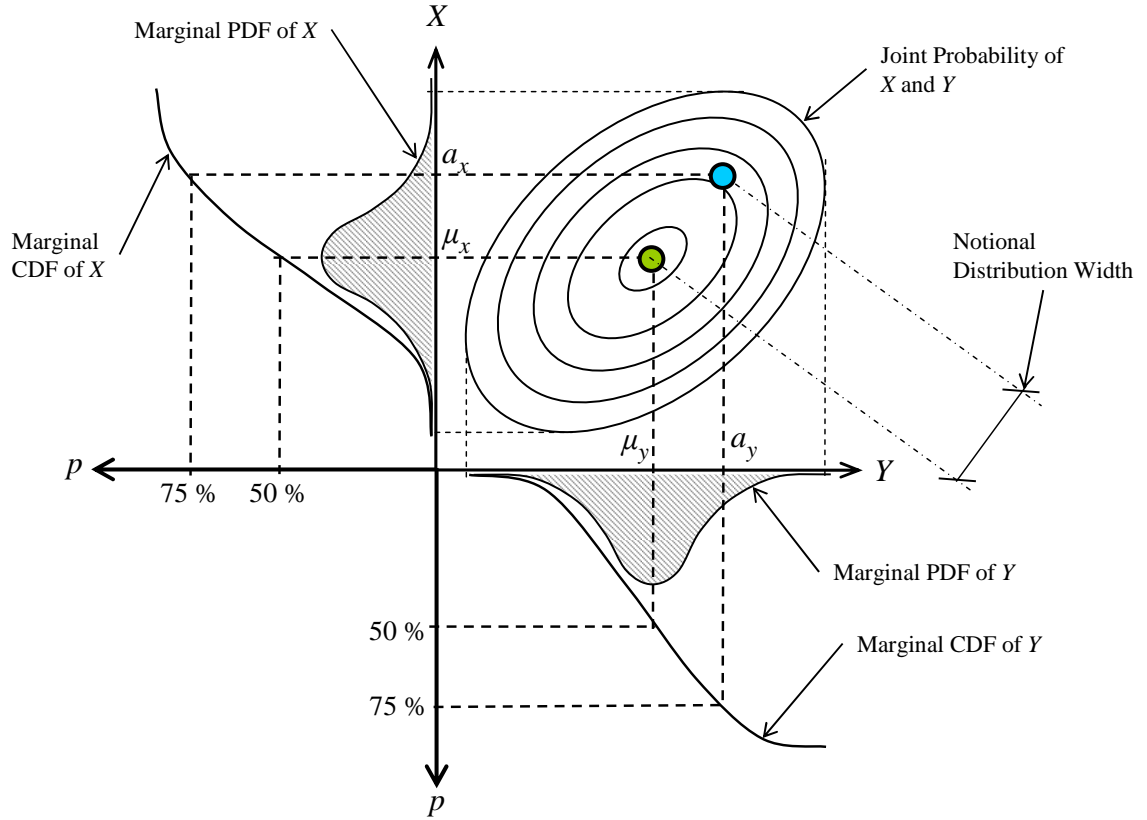


Figure 49: Joint and Marginal Probability Distribution

Definition: The **Marginal Distribution** of a random variable X is obtained from the joint probability distribution of X and Y by integrating over the values of random variable Y .

The concept of joint and marginal probability distributions is illustrated in Figure 49 for a two dimensional case. It should be noted that the CDFs considered in the previous section in Figure 48(b) are from the marginal distribution of respective responses.

When the value of one random variable from a pair of jointly distributed random variables is fixed, the distribution on other variable is called conditional distribution. The conditional distribution is a distribution of one random variable conditional to the other taking a particular value.

Definition: For jointly distributed random variable X and Y , the PDF for **Conditional Distribution** of X given $Y = y$ is given by:

$$f_{X|Y=y}(X) = \frac{f(x, y)}{f_Y(y)}$$

where $f_Y(y)$ is the marginal distribution of random variable Y .

In contrast to the conditional distribution, the marginal distribution of X is the distribution on the random variable X when nothing is known about the random variable Y [120].

This discussion leads to the concept of correlation and independence. Correlation is a measure of *linear* dependence and defined by the covariance of two random variables. When it is zero, two random variables are said to be uncorrelated. When two variables are uncorrelated, they are not necessarily independent. Independence is a stronger concept. Random variables X and Y are independent if any function of X is uncorrelated with any function of Y . Thus if two variables are independent they are definitely uncorrelated, but vice versa is not true.

Definition: Two random variables are independent of each other if their joint probability density function is the product of their marginal distributions.

$$f(x, y) = f_X(x)f_Y(y)$$

As a consequence of independence of X and Y , their conditional distributions are identical to their marginal distribution. As an extension of this concept, if the random variables have a multi-variate *normal* distribution and are pairwise uncorrelated, then the random variables are always independent [139]. It should be noted that this is a special property of multi-variate normal distribution and is not true for any other distribution.

From the above discussion, it is clear that accurate representation of uncertainties in a multi-dimensional objective space can only be accomplished via the use of

joint probability distribution. Decisions should be made based on the conditional probabilities of various responses. This idea has been successfully implemented by Bandte [133] for making system design decisions and by Garvey [140] for analyzing program cost and schedule uncertainties. One of the main limitations of using joint probabilities for technology decision making is the scale of the problem. The problem of calculating joint PDFs and conditional probabilities for a large number of technology combinations in a large dimensional objective space is intractable. This cannot be implemented for probabilistically comparing each technology combination for Pareto optimization.

To overcome this limitation, marginal probabilities can be used and integrated within the *a posteriori* preference articulation framework. The mathematical accuracy of this implementation is only guaranteed if the responses are *uncorrelated* and their joint distribution is a *multi-variate normal distribution*. This way, the conditional probability distribution of each response will be its marginal distribution. Thus the main assumption here is that the responses are independent of each other. Even if this is not true, the use of marginal distributions is still theoretically sound, only drawback being that the results will not capture the nuances of conditional probability.

6.5.2 Probabilistic Pareto Layers

Based on this observation, an approach for probabilistic technology selection is proposed where the DM is presented with multiple *layers* of Pareto fronts. This is illustrated in Figure 50 for a notional 2-dimensional case. Here each Pareto layer consists of solutions corresponding to a specified probability level derived from the marginal probability distribution as illustrated in Figure 49. For example, let's select point a from the Pareto layer of 75% p -level in Figure 50. For this point, (a_X, a_Y) are the values corresponding to a 75% probability on the marginal CDFs of X and

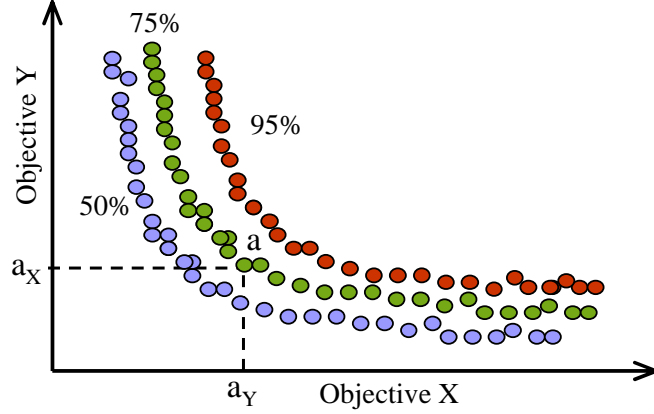


Figure 50: Notional Pareto Frontiers with Different Probability Levels

Y objectives respectively as illustrated in Figure 49.

On the basis of ideas discussed above and in the previous chapters, following hypothesis is proposed addressing one of the primary research questions: *How to account for technological uncertainties while selecting technology combinations?*

Hypothesis: An approach based on probabilistic Pareto *layers* is the most appropriate method of accounting for technological uncertainties within the MODM framework of *a posteriori* preference articulation.

Supporting Experimentation: Soundness of this approach is checked with the help of a benchmark knapsack problem that has variability associated with the items. Results based on deterministic and probabilistic Pareto optimization are evaluated.

There are two primary enablers required for the implementation of this approach. One is the previously discussed probabilistic analysis framework using surrogate models and Monte Carlo simulations. This will be used to calculate the marginal CDFs for each response for each technology combination. The other enabler is *probabilistic Pareto optimization*. This is discussed in the following section.

6.6 Probabilistic Pareto Optimization

The probabilistic Pareto optimization scheme is implemented using the RSM+MCS framework in conjunction with SPEA2. The idea is to obtain desired p -level values (via MCS) for each technology combination of the population of SPEA2 generation. These values are then used to calculate the fitness of the population members. The overall scheme of the algorithm remains same as discussed in Chapter 5. Only the fitness calculating and archiving elements are changed to account for probabilistic evaluations.

6.6.1 Fitness Calculation

Lets take an example where the DMs desire three Pareto layers each representing 50%, 75%, and 95% p -level. Now, for each member of population there will be an $r \times 3$ array of response values with r being the number of responses and 3 for three p -levels. The fitness is calculated for the population in the standard fashion (as discussed in Chapter 5) for each p -level. Thus, instead of having just one fitness value, each population member will have 3 fitness values, one corresponding to each p -level. For the purpose of reproduction operator, the fitness values for all p -levels of a population member are added together; this sum is now used for tournament based reproduction operator.

6.6.2 Archiving

Environmental selection or archiving is one of the most important operators of the algorithm and is considerably modified for implementing the probabilistic approach. There are two main parts of the original archiving operation: *a)* save all the non-dominated points, and *b)* if the number of these points is more than the archive size, implement niching. For the population with multiple p -levels, the modified archive operator retains members with at least one non-dominating p -level value. If the number of retained members is more than the archive size, an archive truncation

procedure is activated that iteratively removes members according to their niching property and distribution width. The distribution width of an individual for this purpose is defined by the Euclidean distance between the largest and smallest p -level values. This is illustrated for a notional two dimensional problem in Figure 49. This quantity can be considered proportional to the width of each marginal PDF normalized over all the responses. It is related to the robustness of the individual. For the archive truncation procedure, two nearest points within any of the Pareto layers are selected and the one with widest distribution is removed from the archive. This way, clustering of the solutions in one region is avoided and less robust solutions are discarded.

6.7 Validating the Approach on a Knapsack Problem

The approach stated in the hypothesis above about making decisions based on probabilistic Pareto layers is validated using a benchmark knapsack problem. To investigate its benefits, this approach is compared with results from the deterministic Pareto based approach.

6.7.1 Probabilistic Knapsack Problem

The benchmark problem is the same as described in previous chapters, except this time all the 16 items have uncertainties associated with them. Only first two dimensions are considered to be minimized to facilitate better visualization and analysis. The item options for the knapsack problem are listed in Table 13. Here each item has an impact on two responses X and Y . For simplicity, the uncertainty in this impact is assumed to be triangularly distributed with minimum, most likely, and maximum possible value; these values for each response variable are listed in the Table 13 under Min, ML, and Max columns respectively. As it can be observed, the distributions are not always symmetric around the most likely values. This has been done to represent the actual distributions on the technology impacts which are rarely symmetric.

Table 13: Probabilistic Knapsack Problem

Item No.	X			Y		
	Min	ML	Max	Min	ML	Max
1	5	7	9	-15	-9	-8
2	1	3	4	-8	-6	0
3	1	2	4	-9	-7	-5
4	4	6	8	-7	-5	-3
5	-5	1	3	-6	-5	-3
6	3	5	7	-6	-4	-2
7	4	10	11	-6	-4	-2
8	1	3	10	-7	-5	1
9	6	8	10	-6	-4	-2
10	2	4	6	-11	-10	-8
11	1	3	5	-9	-7	-5
12	8	10	12	-8	-6	-4
13	5	7	9	-11	-9	-5
14	5	7	12	-6	-4	-2
15	3	5	7	-15	-10	-2
16	2	8	10	-10	-8	-6

Moreover, if all distributions are symmetric, the deterministic results may be similar to the probabilistic results at 50% p -level.

As an example case, to select a solution, it is decided that value for response Y should not be more than -60 and that for response X should not be more than 45 . Thus the, constraints are $X \leq 45$ and $Y \leq -60$; they cannot be violated. Moreover, it is desired to have X value as minimum as possible.

6.7.2 Deterministic Results

The deterministic Pareto optimization is implemented using the most likely (ML) values for X and Y from Table 13. The SPEA2 procedure described in Chapter 5 is used for Pareto optimization in two dimensions. The parameter settings used for the simulation are:

Mutation Rate : 0.05

Crossover Rate : 0.8

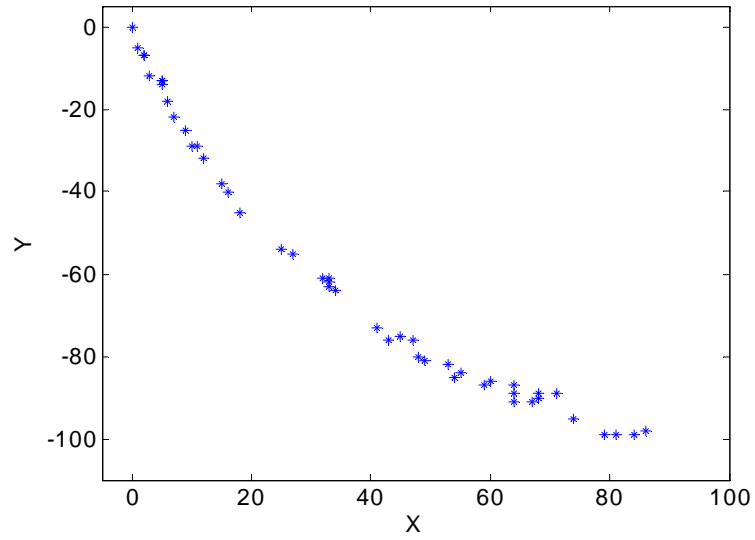
Number of Generations : 100
Population Size : 100
Archive Size : 100

The run time for this simulation is about 12 seconds on a 2 Ghz P4 machine. The Pareto optimal solutions obtained are plotted in Figure 51(a). These solutions are transferred to JMP[®] [53] for further analysis. Based on the problem statement and considering all the constraints, there are 14 feasible solutions on the Pareto front. Out of these, one that has minimum X value, as required by the problem statement, is selected. This point along with all feasible solutions is plotted in Figure 51(b). The response values for this point are $X = 32$ and $Y = -61$. It is obtained by the item combination [1 1 1 0 1 0 0 1 0 1 0 0 1 0 1 0]; this is a 16 bit string where 1 represents the presence of the item and 0 its absence. Thus, 8 out of 16 available items are included in this solution.

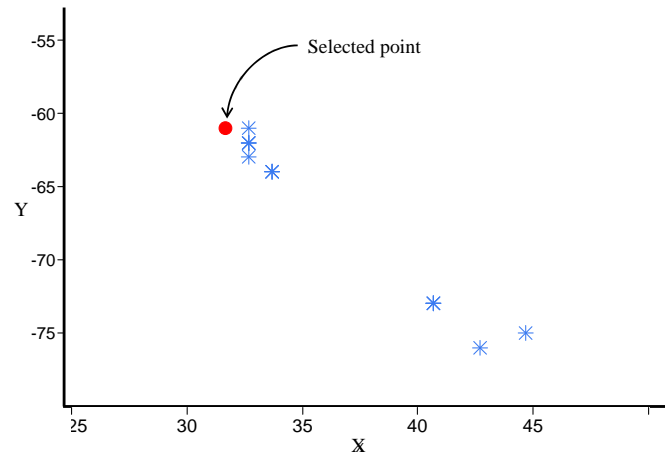
6.7.3 Probabilistic Results

The probabilistic Pareto optimization is implemented using the triangular distributions listed in Table 13. The SPEA2 procedure for finding three Pareto layers of p -values 50%, 80%, and 90% is implemented as described in Section 6.6. The algorithmic parameter settings used here are same as the ones used for the deterministic analysis. The marginal PDFs for the responses of item combinations are calculated using a 500 run Monte Carlo simulation. The time required for this simulation was about 90 seconds.

The Pareto layers obtained as a result of probabilistic Pareto optimization are plotted in Figure 52(a). Imposing constraints on these solutions, there are 20 feasible points in the Pareto layer (PL) for 50% p -level, 14 in 80% PL, and 9 solutions in 90% PL. Out of these points, three points (one for each PL) are selected based on the criteria from the problem statement. These points are plotted in Figure 52(b).



(a) Pareto Optimal Solutions



(b) Pareto Optimal Solutions With Imposed Constraints

Figure 51: Deterministic Pareto Optimization

Table 14: Solutions from Pareto Layers

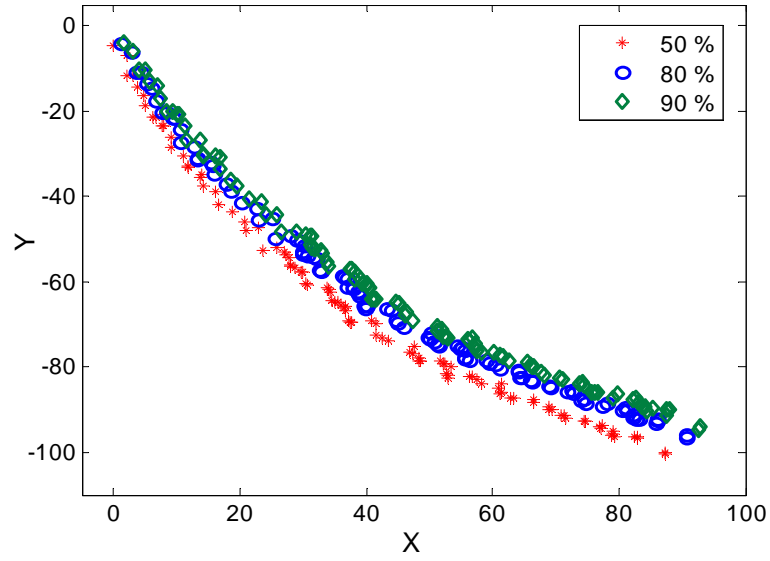
p -level %	Item Combination	X	Y
50	1 1 1 0 1 0 0 0 0 1 1 0 0 0 1 1	30.35	-60.42
80	1 0 1 0 1 0 0 0 0 1 1 0 1 0 1 1	36.97	-61.25
90	1 1 1 0 1 1 0 0 0 1 1 0 0 0 1 1	39.47	-60.32

The three selected points with their item combination are listed in Table 14. It is interesting to note that all three points correspond to three different item combination. This indicates that different item combinations provide the *best* points in different PL. In other words, item combination selected on one PL layer with certain tradeoffs may not correspond to the point on other PL, even if selected based on the same tradeoff criteria.

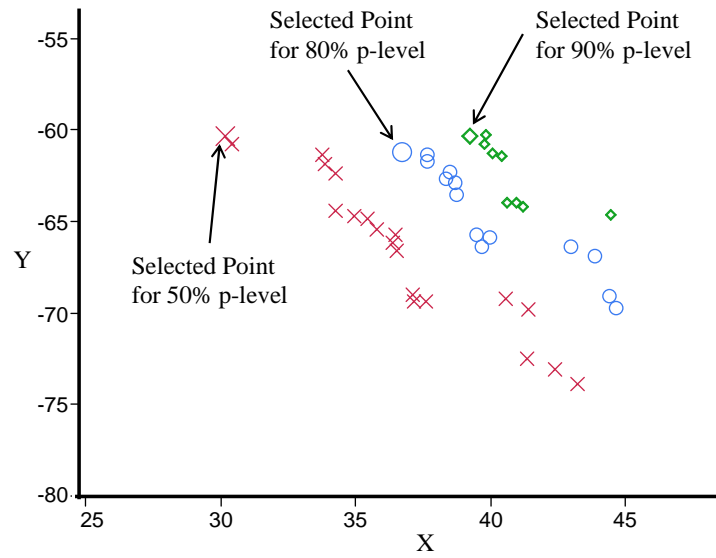
6.7.4 Result Comparison

In order to demonstrate the value of the new approach using probabilistic Pareto layers, its results are compared to the solution selected based on deterministic Pareto optimization. As a first step in this direction, a probabilistic analysis is carried out on the deterministic solution using a 1000 run Monte Carlo simulation. Marginal CDFs for response X and Y are created from the Monte Carlo results and are plotted in Figure 53. One of the main constraints on the solutions is that $Y \leq -60$. It is clear from the CDF plotted in Figure 53(b) that the probability of meeting this constraint is only 30%. Thus the deterministic evaluation leads to a faulty solution.

To compare the three probabilistic solutions with the deterministic, 1000 run MCS is also carried out for the probabilistic solutions. The CDFs for all four solutions are plotted in Figure 54. When considering response variable X , it can be observed from the Figure 54(a) that the deterministic solution would provide good results consistently over all p -levels. It is within the constraint $X \leq 45$ with 100% probability. Only the item combination selected from 50% PL has better values than the deterministic solution. Moreover, all four solutions lie comfortably within the X constraint

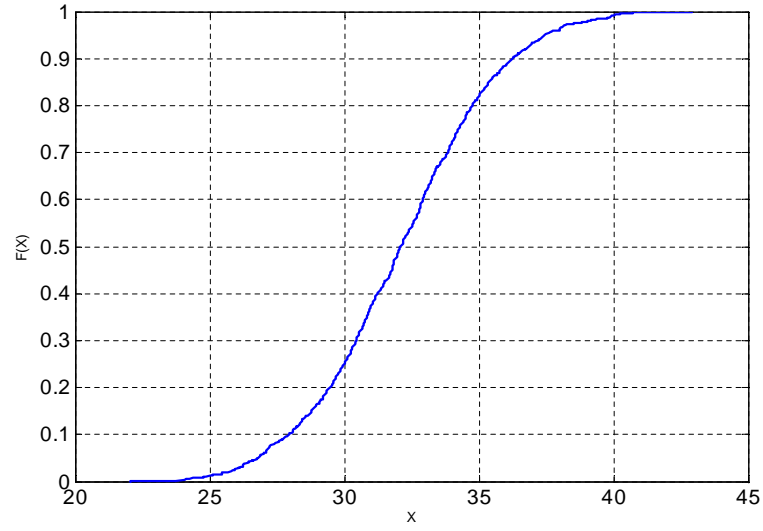


(a) Probabilistic Pareto Layers

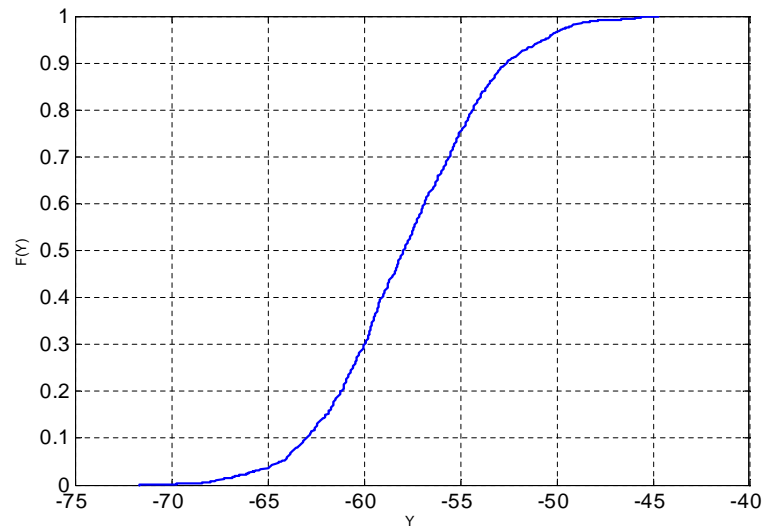


(b) Solutions on Pareto Layers With Imposed Constraints

Figure 52: Probabilistic Pareto Optimization



(a) Empirical CDF for Variable X

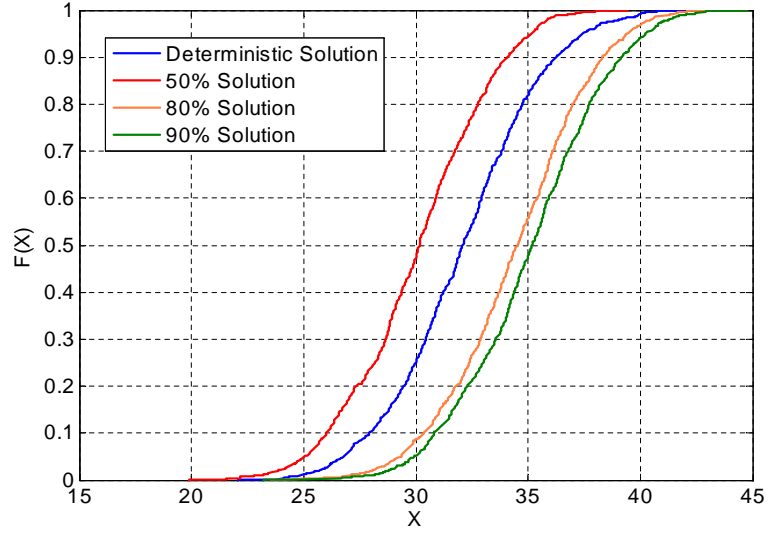


(b) Empirical CDF for Variable Y

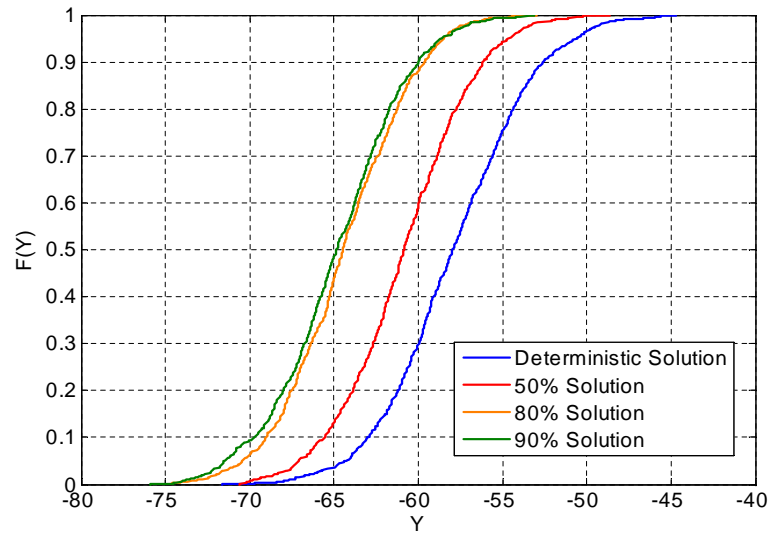
Figure 53: Empirical CDFs for Deterministic Solution

as they are selected from a region very far from the constraint. Comparing the CDFs for response variable Y in Figure 54(b) is more interesting as $Y \leq -60$ can be considered as an active constraint. The points selected are nearest to this constraint in the feasible design space. It can be observed from these CDFs that as one moves up along the CDF, starting with the deterministic solution, the solution becomes infeasible with increasing p -level. Thus, up till 30% p -level the deterministic solution is good. At this point, its Y value increases beyond the constraint and it becomes infeasible. Now, one has switch to the next CDF of the item combination from 50% PL and this solution is feasible up till 60% p -level. The CDFs of the last two item combinations from 80% and 90% PLs are very close to each other. The constraint limit is crossed just before and just after the 90% mark for both solutions. Thus the solution from 80% PL is not feasible for 90% p -level.

From the above example, it can be noted that if a deterministic analysis is conducted, the solution may not be what one expected when the uncertainties are considered. It may even be infeasible. In such situations, the constraints have to be relaxed, which may not be an option in many design problems, for example when emissions and noise constraints corresponding to the government regulations are present. As a result, another solution has to be selected and probabilistic analysis conducted till one finds a feasible solution with required p -level. Now, on the other hand, if the technology combination is selected from a Pareto layer of required p -level, it is guaranteed to satisfy all the requirements and post-optimality probabilistic analysis is not required. Thus making technology decisions based on *a posteriori* preference articulation framework with probabilistic Pareto optimization would eliminate the iterative step mentioned before. Moreover, this approach can also provide technology combinations that are not Pareto optimal with deterministic evaluation but are better solutions when uncertainties are considered.



(a) Empirical CDF for Variable X



(b) Empirical CDF for Variable Y

Figure 54: Empirical CDFs for Deterministic and Probabilistic Solutions

6.8 Summary

This chapter has focused on the technological uncertainties and how to account for them while selecting technologies for a complex system. The field of probabilistic design was investigated for this purpose. Three main ingredients involved in probabilistic design are identified: *uncertainty quantification*, *probabilistic analysis* and *probabilistic optimization*. The use of beta distributions created based on expert opinion is advocated for representing epistemic uncertainties in technology impacts. Various probabilistic analysis techniques are described and one based on response surface equations and Monte Carlo simulations (MCS) is considered appropriate for the purpose. Different probabilistic optimization techniques are discussed. None of the existing techniques is found appropriate for the current application. Thus, a probabilistic Pareto optimization technique is proposed. The idea here is to present Pareto layers of different probability levels to the decision makers. The decision makers can make tradeoffs among various objectives and also probability levels to select a satisficing technology combination. This idea leads towards some questions regarding the nature of distributions in a multi-dimensional space. Joint probability distribution is investigated in this regards. The marginal distribution of each response in a multi-dimensional space is considered appropriate for calculating the probability values.

It is hypothesized that the new technique is better for probabilistic decision making than just considering a deterministic Pareto front and then probabilistically analyzing the selected solution. For implementing this technique, the evolutionary algorithm for Pareto optimization is modified to be used in conjunction with MCS. MCS provides the marginal distribution for each response. The fitness and archive functions of the Pareto EA are modified to handle probabilistic values. The plausibility of the hypothesis is checked with the help of a benchmark knapsack problem. The results obtained from deterministic Pareto front and probabilistic Pareto layers are compared. It is

observed that given the same problem statement and in the presence of uncertainties, selected solutions from probabilistic Pareto layers will always be better and more realistic in terms of satisfying constraints and other requirements, than the one selected from a deterministic Pareto front. Thus the plausibility of the hypothesis on using probabilistic Pareto layers for Pareto optimization is verified.

CHAPTER VII

TECHNOLOGY CONSTRAINTS

Analysis of technology constraints is an important aspect of a technology selection process for large scale complex systems. Technologies can interact with each other in a variety of ways and are manifested in the form of their impact on the system. It is important to ensure that there are no conflicting or incompatible technologies present in the group of selected technologies. This is a combinatorial optimization problem where the problem size geometrically increases with the increase in the number of technology options available. Technology interactions act as a constraint in this combinatorial optimization and tend to reduce the total number of permissible combinations and at the same time making the entire search space more complex.

This chapter will discuss some of the intricacies involved with technology constraints with primary focus on technology incompatibilities. Some techniques used to account for them are discussed and a new approach to analyze technology constraints based on the principles of *Graph Theory* is introduced. A new metric to quantify the computational complexity of the technology combinatorial space is presented. As a result of insights gained from this study, an approach to account for technology constraints within the Pareto optimization framework is selected.

7.1 Types of Technology Interactions

Various types of interactions or relations exists among technologies. An initial attempt to model technology interactions in the context of technology selection for preliminary aircraft design is described by Kirby [13]. In this treatment of interactions, physical compatibility/incompatibility rules between technologies are formalized in

the form of a *Technology Compatibility Matrix* (TCM). Roth and Patel [141] categorize various types of interactions that exist among technologies into two main groups: *Simple Interactions* and *Non-Simple Interactions*.

7.1.1 Simple Interactions

Simple technology interactions are boolean relationships among technologies. The basic types of boolean technology interactions are shown in Figure 55 (adapted from [142]). The most likely relationship that exists among technologies is of independence. That is a technology is completely independent of the rest and can be used with any other technology. In other words, it is compatible with all the technologies and does not interact with any other. The next is incompatibility, where a technology is not compatible with another and the two cannot be used together. Hence, either technology a OR b has to be used. Incompatibilities arise when two technologies are competing for the same function or when one technology severely degrades the functionality of another. For example, there can be two structural technologies such as composites and integrally stiffened aluminium for construction of wings and only one can be used. As this relationship is symmetric it can be accounted by using only the super diagonal elements of a $n \times n$ (square) matrix as shown in Equation 33. Here, for any i, j such that $1 \leq i \leq j \leq n$, if technology i and j are incompatible, then $c_{i,j} = 1$, otherwise 0.

$$\mathcal{C} = \begin{pmatrix} 0 & c_{1,2} & \dots & c_{1,n} \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & 0 & c_{n-1,n} \\ 0 & 0 & \dots & 0 \end{pmatrix} \quad (33)$$

Another form of boolean interaction that can be present among technologies is an *Enabling* relationship. Here, the presence of one technology is necessary for proper functioning of the other, therefore, technologies a AND b have to be used together.

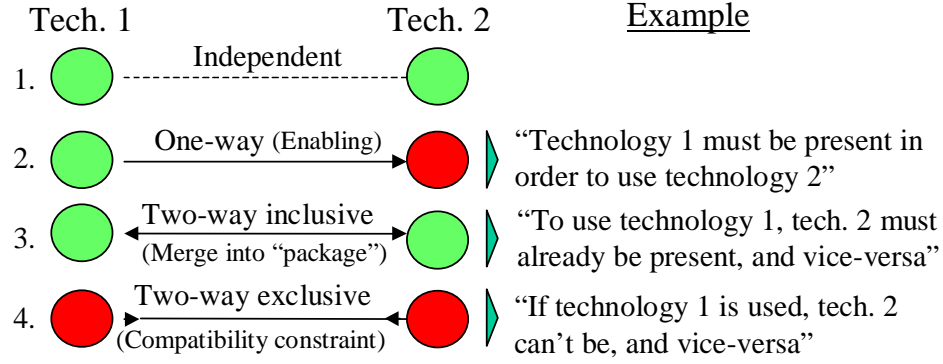


Figure 55: Simple Technology Interactions

Enabling relationship is not symmetric and can act in two directions. Either a can be an enabling technology for b i.e. a can work independently while b cannot work without a , or vice versa. There can also be a much stronger relationship where neither a nor b can work independently. In this case these two technologies can be merged into a “package”. For enabling interactions, as the relationship is not symmetric, both the sub and super diagonal elements of a $n \times n$ matrix are required to define the interactions as shown in Equation 34. In this formulation, for any i, j such that $1 \leq i, j \leq n$, if i is an enabling technology for j and i is independent of j then $e_{i,j} = 0$ and $e_{j,i} = 1$.¹ If both i and j are enabled by each other then $e_{i,j} = 1$ and $e_{j,i} = 1$.

$$\mathcal{E} = \begin{pmatrix} 0 & e_{1,2} & \dots & e_{1,n} \\ e_{2,1} & 0 & \ddots & \vdots \\ \vdots & \ddots & 0 & e_{n-1,n} \\ e_{n,1} & \dots & e_{n,n-1} & 0 \end{pmatrix} \quad (34)$$

Boolean relationships are the most common form of interactions that exist among a pool of technology options for modern complex system.

¹ $e_{i,j}$ is read as i is enabled by j .

Table 15: Technology Constraint Matrix

	T1	T2	T3	T4	...
T1	0	1	0	-1	
T2	0	0	-1	0	
T3	0	0	0	0	
T4	-1	0	0	0	
\vdots					\ddots

7.1.1.1 Technology Constraint Matrix

While implementing simple technology interactions in the TIES methodology, the compatibility relationship in form of Equation 33 and enabling relations in form of Equation 34 are combined in a *Technology Constraint Matrix*. It is possible to combine the two equations into one because the two relationships are mutually exclusive. That is to say that when two technologies are incompatible, they cannot be enabling each other at the same time and vice versa. Here, the enabling technology relationship is denoted by “-1” instead of “1” as it conflicts with the notation of incompatibility relationship. A notional technology constraint matrix is listed in Table 15.

7.1.2 Non-Simple Interactions

Simple technology interactions as described before are primarily boolean relationships. Here, the impact of technology interactions on system level metrics is additive. That is when two technologies enable each other and are considered together for a technology combination, their combined impact on a system level metric is the sum of each technology considered individually. When the technologies are incompatible, only one technology can be considered at a time. This assumption is a vast simplification and generally not valid when real cases are considered. It considerably limits the technology combinatorial space. There can be various levels of interactions between two technologies rather than just -1, 0 and 1 as denoted in TCM. These type of interactions are called non-boolean interactions [141]. For example, if the impact of

technology T_1 on a certain metric is x and that of technology T_2 on it is y ; when these two technologies are considered together, the total impact may not be $x + y$. It can be some other function of x and y . These type of interactions have to be considered on a case by case basis. For the technology problem under consideration, they are accounted within the technology evaluation model.

Various types of more complex boolean interactions also arise among technologies. A simple example is a three way interaction arising among three technologies. If the technologies are independent, there are 8 permissible combinations. However, if all are incompatible with each other, three technologies can only be used independently or none is used, i.e. 4 permissible combinations. In general, it not easy to count the exact amount of permissible technology combinations. Principles of *Graph Theory* can help us enumerate permissible combinations and better understand the technology combinatorial design space. Graph Theory is an area of discrete mathematics and the relation of technology interactions with this field is explored in following section.

7.2 *Graph Theory Connection*

A *graph* is a triple consisting of a vertex set, $V(G)$, an edge set, $E(G)$ and a relation that associates with each edge two vertices (not necessarily distinct) called its end-points [143]. A graph can be used to represent a technology space, vertices represent the technologies and edges represent the interaction between two *distinct* technologies. For now, we denote non-directional edges between technology vertices and these edges represents incompatibility relations. A notional technology space with compatibility constraints is shown in the form of a graph in Figure 56. Here, T3 and T8 do not have any edges incident to them, hence they are totally independent technologies. However, for example, T1 has two incident edges and therefore it is incompatible with two technologies namely T7 and T4.

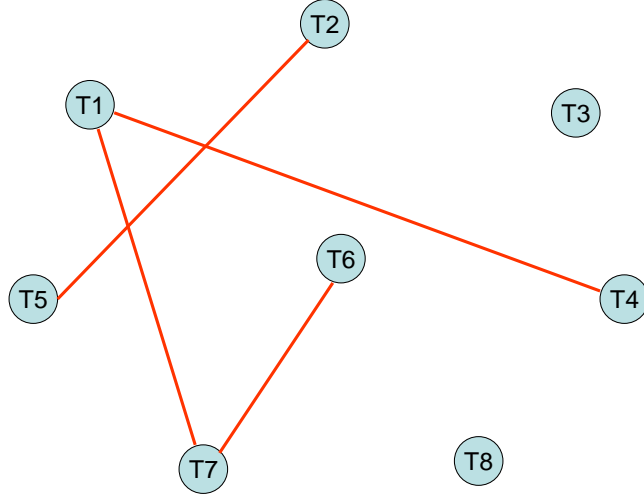


Figure 56: Technology Graph T

7.2.1 Counting Permissible Technology Combinations

Permissible technology combinations are sets of technologies that do not violate any compatibility or enabling constraints. Even considering only the incompatibility constraints, it is difficult to quantify or enumerate the number of permissible technology combinations. Graph theory can help tackle this problem. As mentioned before, the technology space is seen as a graph with technologies as vertices and non-directional edges as incompatibility constraints. The maximum number of edges a graph can have is given by:

$$\binom{n}{2} = \frac{n}{2}(n-1)$$

This is equivalent to the maximum number of incompatibilities a group of technologies can have among themselves, and in such a situation, each technology can be used individually or none at all. Therefore, the maximum number of permissible combinations here will be $n + 1$. When all the technologies are independent and there are no edges between them, the maximum number of permissible combinations is 2^n .

The number of permissible combinations to be counted in the above mentioned

extreme cases is trivial, but it is a difficult problem when the number of incompatibilities is between 0 and $\binom{n}{2}$. In graph theoretic parallels, the problem is to find total number of independent sets. A subset S of $V(G)$ is called an *independent set* of G if no two vertices of S are adjacent² in G [144]. The number of independent sets in T not only depends on the number of vertices and edges but also on the arrangement of edges between the vertices. For example, different arrangements of 10 incompatibilities among 10 technologies that give maximum and minimum number of independent sets possible is shown in Figure 57. The maximum number of independent sets are obtained when one technology is incompatible with all other technologies and the remaining are as independent among themselves as possible. In other words, one vertex has maximum degree³, $n - 1$ in this case, and the remaining vertices have minimum possible degrees. This arrangement is demonstrated in Figure 57(a) and the vertex degrees are $[9, 2, 2, 1, 1, 1, 1, 1, 1, 1]$. On the other hand, minimum number of independent sets are obtained when the technologies form groups or components that are complete graphs in themselves, i.e. all the technologies within a component are incompatible with each other. This arrangement is represented in Figure 57(b) with 3 triangles and the remaining vertex attached to one of the triangles. The above observations are made using an integrated environment for graph theory called *newGRAPH* [145].

While analyzing real technologies, one finds majority of them are independent and the remaining are not completely interconnected but form small components of mutually interacting technologies. This fact can be exploited while calculating the total number of permissible combinations as the problem of enumerating the independent sets of a large connected graph is difficult and computationally intensive. Let us consider a real example with 29 technologies, out of which, 17 are totally independent and 12 technologies have 11 incompatibility constraints among them as

²Two vertices are *adjacent* if there is an edge between them.

³The *degree* of vertex is the number of incident edges

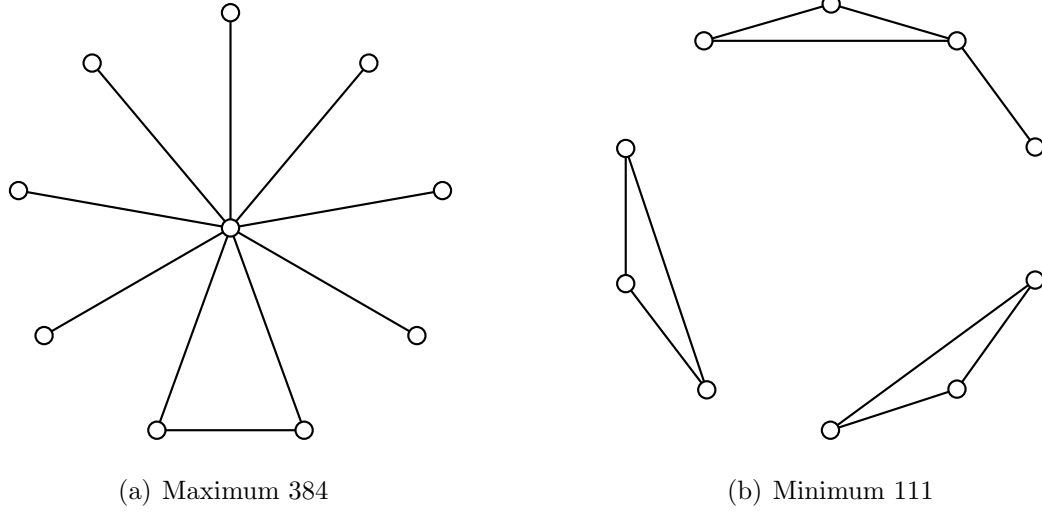


Figure 57: Permissible Combinations with $n = 10$ and $e = 10$

depicted in Figure 58. This graph has four disconnected components. Here each component has a maximum 4 vertices and it is easy to manually count the number of independent sets for each component. Now, let a and b denote two components with i_a and i_b number of independent sets (not counting the null set) respectively. With basic combinatorics, when these two components are included in a single graph, i.e. union of two components, the total number of independent sets of $a + b$ is given by Equation 35.

$$i_{a+b} = i_a \times i_b + i_a + i_b \quad (35)$$

In general, for a graph G with w components, the number of independent sets is given by Equation 36. In many examples, these components are complete graphs or cliques, i.e., each technology is incompatible with every other technology in the component. In such cases, the number of independent sets for the components is same as their cardinality and Equation 36 becomes similar to the one described by Utturwar et al. [33].

$$i_G = \prod_{j=1}^w (i_j + 1) - 1 \quad (36)$$

Now, the number of independent sets along with the null set for a union T of graph

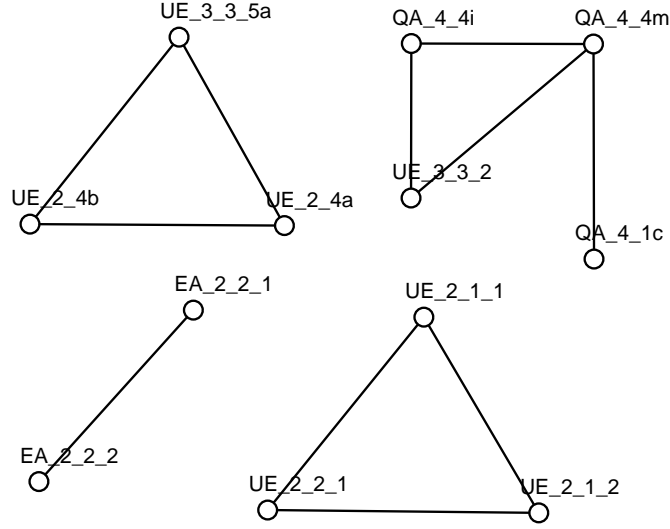


Figure 58: 12 Interacting Technologies From Total of 29

G having i_G independent sets and k independent vertices is given by Equation 37.

$$i_T + 1 = 2^k \times (i_G + 1) \quad (37)$$

Applying Equation 36 for the four components of Figure 58, the number of independent sets is 335. Considering the remaining 17 independent technologies and applying Equation 37, the total number of permissible technology combinations are 44,040,192 (including the null set). This is out of $2^{29} = 536,870,912$ possible combinations. Thus, over 90% of the total technology combinations become impermissible by only about 2.7% of the total possible edges or incompatibilities.

7.2.2 Average Number of Independent Sets

Before investing the time and resources to precisely enumerate permissible combinations, it is useful to know the average number of independent sets a technology graph can have. Random graphs and associated probabilistic techniques are useful for this type of analysis as illustrated by Wilf [146]. Let us consider a random graph $G^p(n, p)$ with n vertices and p is the probability with which each of the $\binom{n}{2}$ edges occur independently. If $S \subseteq V(G^p)$, then the average number of independent sets is the sum of

the probability that every S is independent, over all the vertex subsets S . If S has m vertices, then the probability that S is independent is same as the probability that there are no edges among m vertices of S . With $(1 - p)$ probability of absence of edge between two vertices and $m(m - 1)/2$ edges possible in S , the expression for average number of independent sets is given by Equation 38.

$$I_{G^p} = \sum_{m=0}^n \binom{n}{m} (1 - p)^{m(m-1)/2} \quad (38)$$

For the notional example with 10 technologies and 10 incompatibilities or edges of Figure 57, the fraction of edges present out of total possible 45 is $10/45 = 0.2222$. Applying Equation 38 with $n = 10$ and $p = 0.2222$ we get $I_{G^p} = 174.88$. This number is closer to the lowest possible value of 111 than the maximum number of 384 because there are more arrangements of edges on a random graph that result in the values closer to the minimum than the ones that result in the values closer to the maximum. For the example with 29 technologies and 11 edges, $I_{T^p} = 5.2 \times 10^7$ and the actual number of combinations as counted in the previous section is about 4.4×10^7 . Thus, whenever the technologies interact within small groups and these groups are almost complete graphs, the number of permissible combinations can be significantly lower than the average number of independent sets of corresponding random graph.

7.3 Enumeration with Backtracking

Previous results show that the average number of independent sets can be considerably smaller than 2^t for certain types of technology graphs. This average number give an upper bound for the number of permissible technology combinations. Hence, if I_{T^p} is within the limits of available computation resources, it may be feasible to analyze of all permissible combinations and extract the *true* Pareto optimal solution set, instead of going with a stochastic optimization approach which is approximate in nature. Now, to evaluate all permissible technology combinations, it is necessary to enumerate them. A prevalent search technique called *backtracking* is described that

can be used to enumerate the permissible combinations. This technique is generally used to solve graph theoretic problems such as finding maximum independent set or clique [147], graph coloring, etc. Backtracking essentially performs a depth first search on the technology graph.

Consider a graph G with 6 vertices and 7 edges as shown in Figure 59. Starting with the first vertex, the independent set is $S := \{T1\}$. Now, we attempt to enlarge S and the next vertex we can add is T3 as T2 is connected to T1. The S now has $\{T1, T3\}$. After T3 we can only add T6 and cannot go any further, S is $\{T1, T3, T6\}$. Therefore, we backtrack one step at a time till we can find more options. In this example, we have to go back to T1 (delete T3 and T6 from S) and search for the next vertex that can be added, here it is T5. When all options are exhausted with T1, we start the process again with the next vertex and $S := \{T2\}$. A list of independent sets for the example as obtained by backtracking method is enumerated below.

$\{T1\}, \{T1, T3\}, \{T1, T3, T6\}, \{T1, T5\}, \{T1, T5, T6\}, \{T1, T6\}$

$\{T2\}, \{T2, T4\}, \{T2, T5\}, \{T2, T5, T6\}, \{T2, T6\}$

$\{T3\}, \{T3, T6\}$

$\{T4\}$

$\{T5\}, \{T5, T6\}$

$\{T6\}$

As observed before, the technology space for real problems is composed of small disjoint components and other independent technologies. Independent sets in each of these components can be enumerated using backtracking technique. In the technology evaluation environment of described in the previous chapters, a technology combination is represented by a row vector of zeros and ones; for e.g., a combination of T1, T3 and T6 in a graph with 6 technologies is represented as $[1, 0, 1, 0, 0, 1]$. A

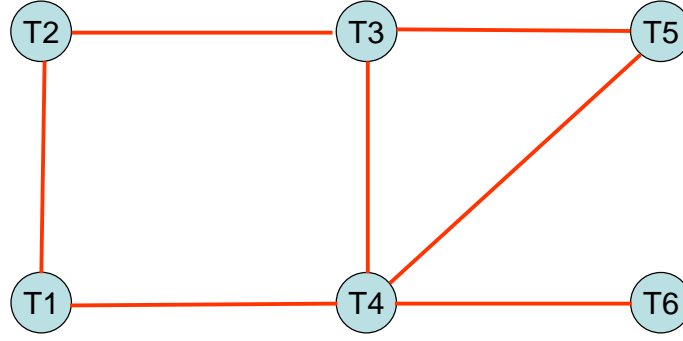


Figure 59: Graph G for backtracking

set of all permissible combinations in a component with n technologies is in the form a $i \times n$ matrix, with each row representing an independent set. The matrix of permissible combinations for n independent technologies is basically a binary conversion of a row of numbers from 0 to $2^n - 1$, with 2^n rows and n columns. Now, with matrices of permissible sets for all the components and independent technologies in place, the independent sets of the entire technology graph are enumerated using the logic behind Equation 36 and 37. Consider two components a and b with independent set matrices of size $i_a \times j$ and $i_b \times k$ respectively. The independent sets for the union of a and b are obtained by concatenating each row of the first matrix with each of the other. This will result in a matrix of size $(i_a \times i_b) \times (j + k)$. This process is repeated till all the components and independent technologies are included.

7.4 *Enabling Technologies*

Observations made in previous sections consider only the incompatibilities in the technology space. There may be some technologies in the space that enable others and these can be visualized using graphs with directed edges known as digraphs as shown in Figure 60. Here, the edges point towards the enabling technology, e.g., in Figure 60, T1 is enabled by T3 and T3 is enabled by T2. Hence, while T2 can function independently, T1 needs T3 and T3 needs T2 to function. Depending on

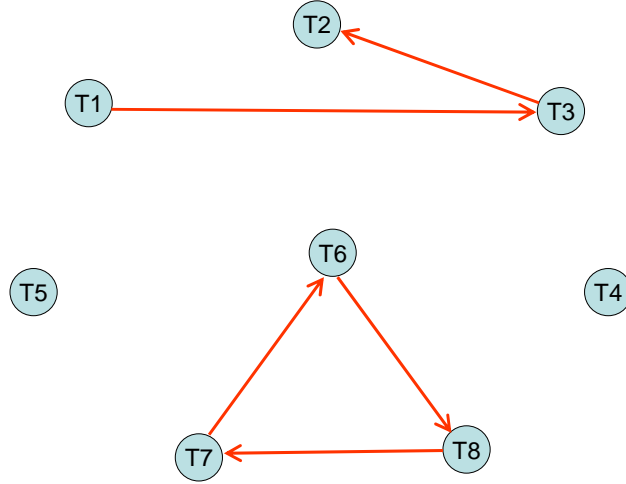


Figure 60: Digraph for Enabling Technologies

the relationships, the complexity of this digraph may be reduced by merging some of the technologies. In Figure 60, T6, T8 and T7 form a unidirectional cycle where one technology is enabled by the next. These can be merged into a single technology as no one can function in absence of any other member of the cycle. This reduction can be adopted for any number of technologies as long as they form a unidirectional cycle and also for two mutually enabling technologies. Once the technology graph is reduced, backtracking technique can be applied with appropriate modifications to account for enabling relationships to enumerate the permissible combinations.

7.5 Technology Constraints with Evolutionary Algorithms

When the number of permissible technologies is too large for a complete evaluation of the combinatorial space, an EA based approach is recommended for Pareto optimization. Two basic approaches have been developed in last decade to account for interactions while using evolutionary algorithms (EAs) for technology selection process.

7.5.1 Soft Constraints

This approach is a type of penalty method where the technology incompatibilities are treated as an objective function whose value is to be reduced through the generations of EA. Here, the incompatible technology sets may also be evaluated. This technique is employed by Roth and Patel [141] where incompatibility free final solution set were obtained with high enough weighting on the incompatibility constraints. The only information needed for this technique is the number of incompatibilities and enabling constraints present in certain set of technologies and there is no need to name the edges that cause those constraints. This number can be easily evaluated using adjacency matrix of the technology graph. For this, two different matrices are created, one for incompatibilities and one for enabling. The adjacency matrix for technology graph with non-directional edges representing incompatibilities is a symmetric matrix where the $(j, k)^{th}$ entry represent the presence or absence of edge between vertices j and k . Matrix C of Equation 33 is the upper triangular portion of the adjacency matrix. When the technology combination set is in the form of a $(1 \times t)$ vector S as shown before, it can be easily proved by basic algebra that the quantity $S \times C \times S^T$ gives the number of edges present in the technology set S .⁴ For evaluating the number of enabling violations in S , adjacency matrix for the digraph is considered and this is same as matrix E of Equation 34. In this case we are interested in the absence of directed edges in S and the number of enabling violations is given by the expression $S \times C \times \bar{S}^T$; here \bar{S} denotes the vector S with all the ones changed to zeros and zeros to ones. If S is a $(n \times t)$ matrix for n technology combinations, above expressions can be used and the result of the product is a $n \times n$ matrix. The number of constraint violations for n combinations are found in n diagonal elements of the resultant matrix.

⁴ S^T denotes transpose of S

This technique for accounting interactions is very simple to implement with EAs. Its main drawback is that there is some probability that the final solution set has incompatible combinations. In case of Pareto optimization, the algorithm has to keep track of extra responses which will have a degrading effect on its performance. Moreover, as observed before, almost 90% of total combinations are impermissible in a technology problem. Given this high proportion of incompatible combinations, the populations for initial generations in SPEA2 will have a very small number of useful combinations. This would severely hamper the performance of the algorithm.

7.5.2 Hard Constraints

In this approach, the technology combinations that violate the incompatibility and enabling constraints are never included in the population pool of the EA. Raczynski et al. [148] proposed a gene correction technique that allows only the compatible technology combinations to be evaluated by the optimizer. This algorithm detects incompatibilities in a technology set and removes certain technologies randomly from the set so that the resulting combination has all compatible technologies. This algorithm can be extended to search and repair for enabling technology combinations. It is included in the EA loop just before the fitness evaluation operator so that no incompatible combination is evaluated. This technique has been shown to result in early convergence of function values as compared to the penalty method. It is flexible enough to be implemented for any type of technology graph. This technique can be easily implemented within SPEA2 for Pareto optimization.

The next technique that implements the hard constraint approach is the reduced bit system employed by Raczynski et al. [149]. When a certain group of n technologies form a clique or complete graph among themselves, instead of 2^n combinations only $n + 1$ can be used. Thus rather than using n columns or bits to represent n technologies, only $\lceil (\ln(n + 1)/\ln(2)) \rceil$ bits may be used. Thus each combination of reduced

bit system corresponds only to a compatible technology combination. When implemented for all the components of technology graph, reduced bit system eliminates the risk of creation of invalid combinations by the mutation and crossover operators of EA. This is an interesting technique with limited applicability. It can only be implemented where the technology space is divided into cliques.

7.6 *Summary*

It has been observed from this study that technology compatibility constraints have significant effect on the technology combinatorial space. In one of the examples illustrated, 90% of the total combinations become impermissible because of less than 3% of incompatibilities. The principles of graph theory were shown to be very useful for analyzing incompatibilities and resultant technology combinatorial space. Technologies and constraints among them are analogous to the vertices and edges of graphs which are good visualizing tool for technology space. Random graphs provide an important result that gives an upper bound on the number of permissible technology combinations present in the technology space. Based on this number, it can be determined if its prudent to go ahead with evaluating all permissible combinations to find the *true* Pareto front in the combinatorial design space. If complete evaluation is to be carried out, backtracking technique can be used to enumerate all the permissible combinations. A technique for accounting compatibility and enabling constraints within Pareto optimization framework was also described.

CHAPTER VIII

PARETO OPTIMIZATION AND SELECTION OF TECHNOLOGIES

The limitations of methods and algorithms employed in common practice for technology selection were realized after Chapters 2 and 3. These approaches do not address the requirements stated in the research goals of this thesis. Thus, based on the discussions in previous chapters, a method is devised to explore the combinatorial technology space and make informed decisions on selecting technologies for complex systems. This method is called *Pareto Optimization and Selection of Technologies* or *POST* for short. This chapter explains the flow of the POST method.

8.1 *Proposed Method*

In the previous chapters, the multi-objective technology selection problem has been decomposed into two basic themes – decision making in multi-dimensional combinatorial technology space and making these decisions in the presence of technological uncertainties. *A posteriori* preference articulation approach has been suggested to address the basic requirement of multi-objective decision making. A subset of Pareto optimal solutions is required to implement this approach. A stochastic algorithm known as SPEA2 was demonstrated to be most effective for Pareto optimization. Uncertainties are quantified for individual responses in a multi-dimensional space in the form of their marginal probabilities. Techniques for evaluating computational complexity of the problem and reducing the dimensionality are also suggested. All these elements come together in Pareto Optimization and Selection of Technologies

or POST. This is a method to systematically explore various technology combinations available for designing a new system and make informed decisions based on the objectives and uncertainties involved. The flow of this method is illustrated in Figure 61. To be precise, this method can be called Probabilistic Pareto Optimization and Selection of Technologies or P-POST; but, in the interest of brevity, POST refers to the probabilistic approach unless stated otherwise.

The process is designed for efficiency and efficacy of decision making. An attempt is made to reduce the time required on part of the decision makers (DMs) to explore and select technologies, and at the same time, make those decisions based on accurate information. To satisfy these conditions, the process is divided into three distinct phases depending on the personnel involved:

Problem Definition: This is the phase where system designers and technology experts participate. Here, a reference system is defined and decisions regarding the use of high fidelity system models or fast executing surrogate models is made depending on the available information, computational resources, and time. The technologies are also identified at this stage. These are the ones the technologists are currently working on or are available to them, or some Commercial Off-The-Shelf (COTS) technologies that the experts deem appropriate for the system. The technology metrics or k -factors are defined at this stage and the impact of different technologies on them is mapped using a technology impact matrix (TIM). The uncertainty distributions on technology impacts are defined where necessary. The compatibility and enabling constraints among the available technologies are also defined. This phase, though very important, is not the primary focus of this thesis and more on this has been explained by Kirby [13] and Mavris [15] among others.

Pareto Optimization: Technology and system analysts are to be involved in this

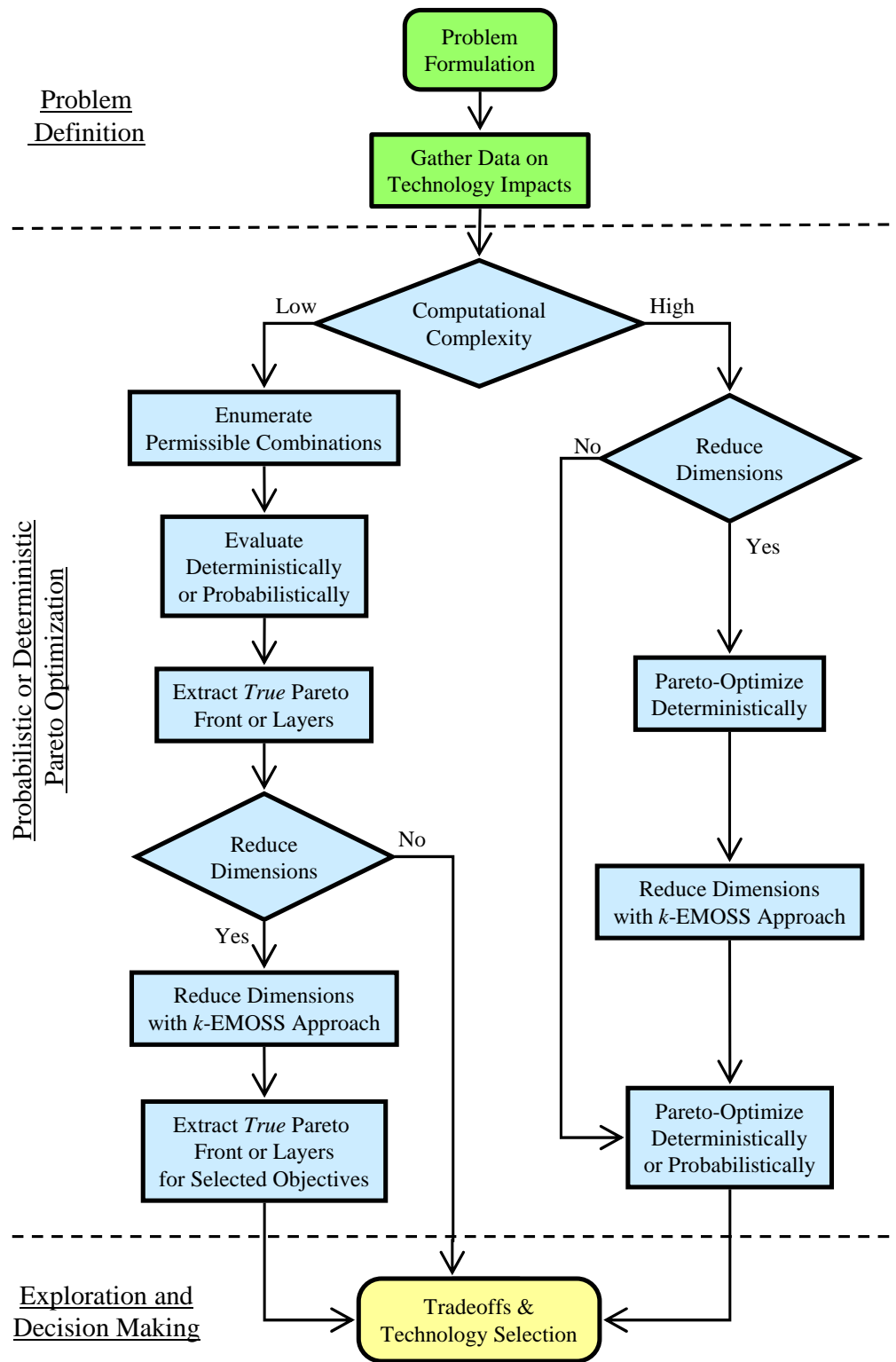


Figure 61: Pareto Optimization and Selection of Technologies

phase along with system designers. The main goal here is of Pareto optimization, probabilistic if uncertainties are present and deterministic if not. Based on the data from the previous phase, this phase generates multi-dimensional Pareto front or probabilistic Pareto layers for the desired probability levels and objectives of interest. This phase is the main focus of this thesis and each of its elements is explained in detail in the following sections.

Decision Making: Once the Pareto front or layers are created, the data is transferred to a selection and tradeoff environment. System designers and other high level decision makers (DMs) are the principle participants in this exercise. In this environment, the DMs have the entire efficient solution space in front of them. They can make implicit tradeoffs among various objectives, compare different solutions deterministically or probabilistically and select the most appropriate technology combination. This multi-dimensional visualization and analysis environment is facilitated by a software from SAS Institute called JMP[®] [53].

There are many steps involved in these phases, especially for optimization. The following sections discuss these steps that form the backbone of the POST method.

8.2 Problem Formulation

While developing a new system, traditional designs may not be able to satisfy the requirements or meet the constraints. To remedy such a situation, as Kirby [13] points out, there are a few options available with the designers. One is to increase the range of design variables and potentially capture a feasible and viable solution; though this may not be possible if the design space was accurately defined initially. The other option is to relax the constraints that are being violated. But, this may not be possible either when the constraints under consideration are non-negotiable constraints, such as government regulations regarding emissions and noise. Now,

with the assumption that the system concept is fixed, the most promising option to design a feasible and viable system is to infuse new and advanced technologies into the system. Thus, the problem of technology exploration and selection is created. At this stage, the system responses based on which the technology decisions are to be based have already been fixed by the designers. These responses are generally some performance, environmental, and economic parameters pertinent to the design. For example, in the case of aircraft design, these responses may include specific fuel consumption, range, weight, takeoff noise, nitrous oxide emissions, acquisition cost, operating cost, etc.

The technologies are evaluated and compared on a baseline system. Thus, selection of this baseline system is also an important part of the problem formulation. Generally, a state of the art system configuration is preferred as a baseline for evaluating the technologies. This is because the simulation models for such systems are powerful enough to account for advance technologies. The design space addressed by older system models may not be large enough to encompass the capabilities of newer technologies.

After deciding on a baseline, various computational models have to be identified for simulating the system. As the response parameters considered for this type of technology exploration exercises are multi-disciplinary in nature, multiple models are usually required to represent the system. For the example set of response parameters, there would generally be at least three types of computer models involved, one each for performance, environmental, and economic responses. These models have to be integrated together to form a multi-disciplinary analysis framework. It is also essential for these models to be physics based so as to capture the impact of technologies at their lowest level. Once the baseline is fixed and physics based system models selected, the system level design variables or technology k -factors are identified. These k -factors are usually a subset of the input parameters to the system models. They are selected

based on their impact on the system responses. A detailed description of the k -factor selection process is provided by Kirby [13]. Various technology combinations will be evaluated on the baseline by considering their impacts of the k -factors and comparisons made using the responses under consideration.

It is often the case that the multi-disciplinary analysis framework for complex systems are computationally very expensive. In such situations it becomes infeasible to evaluate large number of technology combinations in reasonable amount of time. To remedy such situations, surrogate models can be employed to speed up the evaluation process. Surrogate models are a mathematical representation of the physics based system models. Though they are an approximation of the real models, the speed gains far outweigh the loss in accuracy. There are various types of surrogate models that can be used for a given application. Some of the example techniques for surrogate modeling are *Response Surface Equations (RSEs)*, *Artificial Neural Networks (ANNs)*, *Kriging*, *Polynomial Chaos*, *Support Vector Machines (SVM)*, and *Gaussian Process (GP)*. The process for generating surrogate models usually starts with sampling the design space with the help of statistical Design of Experiments (DoE) or some other technique. It is important to verify that the design space considered for this exercise is large enough so as to encompass all the technologies at the same time. These design points are then evaluated using the physics based system model. Finally, a surrogate model based on one of the above mentioned techniques is fitted on these points. It is generally a good idea to check the predictivity of these surrogate models on randomly sampled design points and ascertain that it is within acceptable limits. Each surrogate model thus created would take k -factors as its inputs and the output would be one of the response parameters, also known as the dependent variable.

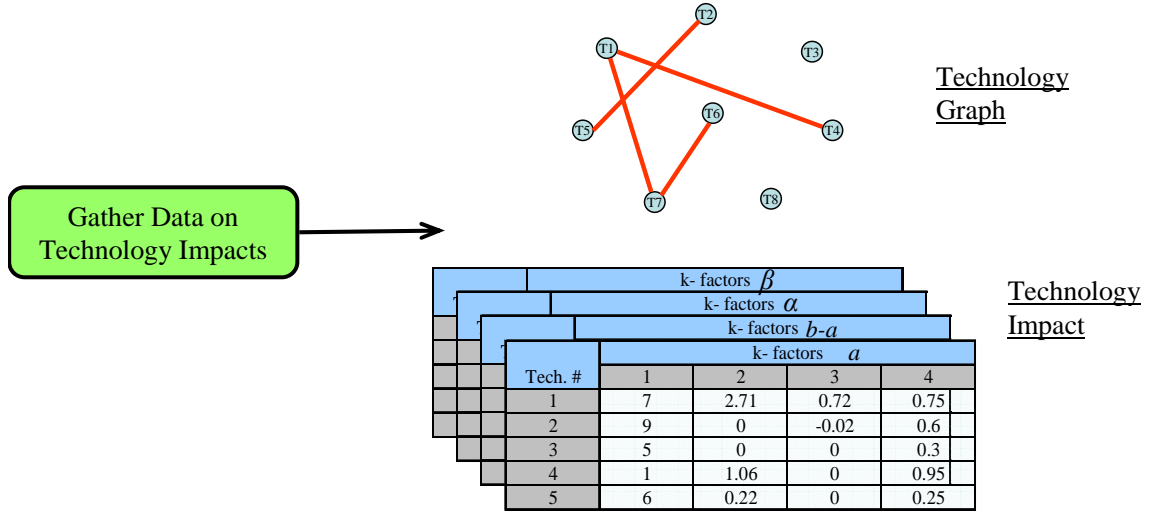


Figure 62: Technology Data

8.3 Collecting Technology Data

This step is part of the first phase of the POST process. The technologies are identified and defined in this step. An illustration of this step with its main outputs is shown in Figure 62. The technology impacts on the k -factors are fixed through a technology impact matrix (TIM). As shown in the figure, TIM is a matrix with rows representing technologies and columns for the impact a technology has on the k -factors. This impact is estimated with respect to the baseline system. The TIM is populated by collecting information from technology and system experts via an audit scheme. For this, a questionnaire is prepared, to gather information regarding technologies and their impact on various k -factors, which is filled in by the experts. Additional information can be gathered from interviews with the technologists and published literature.

If the technology impacts are uncertain in nature, three *pages* of TIM are created to define the minimum (a), most likely, and maximum values (b) of the impact. Based on these three values, four parameters of a generalized beta distribution are calculated for each cell of TIM. These four parameters are location a , scale ($b - a$), α , and β

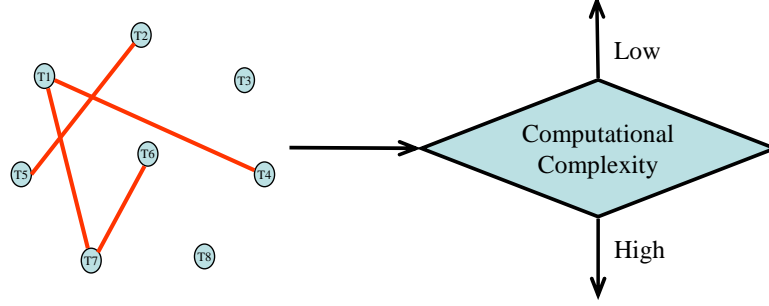


Figure 63: Estimating Complexity of Combinatorial Technology Space

and are calculated based on the formulation described in Chapter 6.

The compatibility and enabling constraints are also defined here and the technology graph is created. This is a graph with nodes representing technologies and edges representing the constraints. Thus there are two main outputs from this step – technology graph and technology impacts; a single TIM if impacts are deterministic and a beta distributed TIM represented by four *pages* for uncertain impacts.

8.4 *Estimating Computational Complexity*

This is the step where the computational complexity of the combinatorial technology selection space is estimated. The main input to this step, as illustrated in Figure 63, is the technology graph created in the previous step. The average number of independent sets or permissible technology combinations is computed for the technology graph from Equation 39 (described in Chapter 7). Based on this average number, the type of analysis to be performed is decided.

$$I_{G^p} = \sum_{m=0}^t \binom{t}{m} (1-p)^{m(m-1)/2} \quad (39)$$

When the average number or permissible combinations is low enough and manageable by the computational resources available, a complete evaluation is carried out to find the *true* Pareto front or layers. For an average desktop with about 2 GHz dual core processor and a 32 bit operating system, if the complexity is within one million

permissible combinations, a complete evaluation can be attempted; given fast surrogate models are available and technology impact is deterministic. If, on the other hand, the average number is high, Pareto optimization via a stochastic algorithm has to be carried out.

8.5 Search for True Pareto Front or Layers

Once it is decided in the previous step that enough computational resources are available, the following steps are carried out for the search of the true Pareto front or layers. Each of the following subsection describes a step under this branch of the POST process.

8.5.1 Enumerate Permissible Combinations

The basic requirement for a complete evaluation and searching for true Pareto layers is the enumeration or identification of all permissible combinations. As described in the previous chapter, the number of permissible combinations is only a small fraction of the total number of combinations. Thus, by enumerating and evaluating only the permissible combinations, considerable amount of computational memory and processor resources can be saved. This enumeration is carried out based on the technology graph previously created. It is implemented using the backtracking and matrix concatenation technique described in Chapter 7. The result of this step is an $n \times t$ matrix, where n = total number of permissible combinations and t = number of technologies. Each row of this matrix is a permissible combination of technologies. For each row, columns with ones represent the presence of corresponding technologies and with zeros represent their absence. The input and output for this step are illustrated in Figure 64.

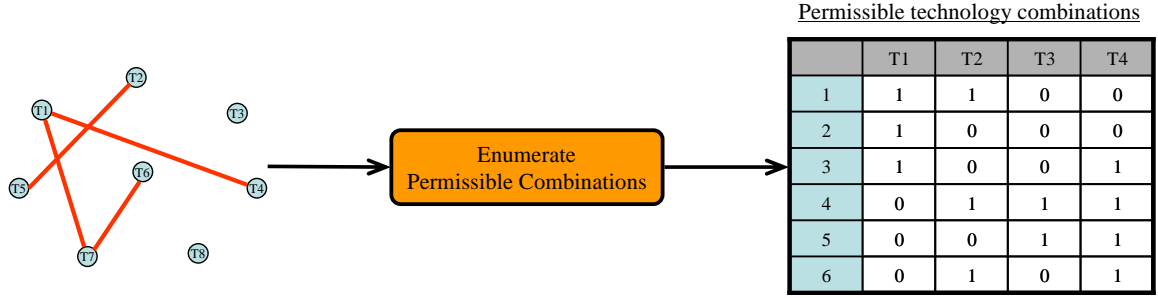


Figure 64: Enumerating Permissible Technology Combinations

8.5.2 Evaluate Deterministically or Probabilistically

The permissible technology combinations enumerated in the previous step are evaluated in this step. If deterministic evaluation is required, physics based system models or surrogate models identified in the problem definition phase are used. For this, the $(n \times t)$ matrix of permissible technology combinations and TIM of size $(t \times k)$ (k being the number of k -factors considered) are first multiplied. This results in an $(n \times k)$ matrix with n vectors of k -factor values for each technology combination. These vectors are then used to evaluate the deterministic impact of each technology combination on the system.

On the other hand, for probabilistic evaluation, the technique using RSM and Monte Carlo simulations (MCS) as described in Chapter 6 is implemented. This is illustrated in Figure 65. For this, each permissible technology combination is subjected to a fixed number of MCS iterations. The beta distributions on technology impacts obtained from the *Technology Data* step are used to generate random samples for MCS. If the number of MC iterations is 500, then 500 random TIM are generated based on the assigned distributions for each TIM element. These TIM are then evaluated in the same manner as done for deterministic analysis. Here, it should be noticed that the computational complexity of the problem significantly increases because of the probabilistic analysis. If there were 100,000 permissible combinations, then for a

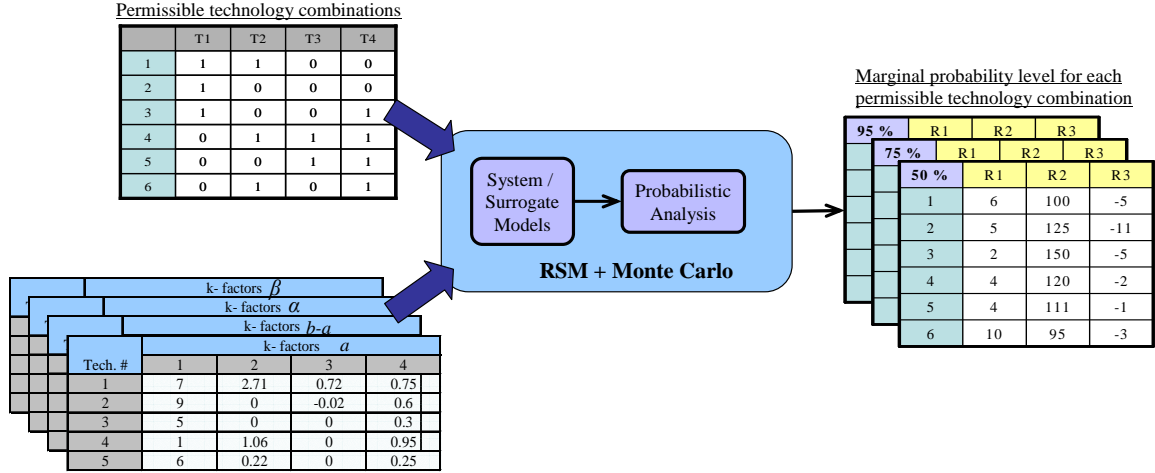


Figure 65: Evaluating all Technology Combinations

500 sampled MCS the total function calls are 50,000,000. That is, 500 times more function calls are required as compared to deterministic analysis.

After the MCS is completed, the response values at predefined probability levels are calculated. These are obtained from the marginal CDFs of each response for each technology combination. Two or more probability levels are considered for this purpose. For probabilistic technology exploration, the decision makers are usually interested in higher probability levels rather than the lower ones. Hence, the probability levels of 50%, 75%, and 95% are well suited for the purpose. The output is in the form of a three dimensional array with rows for each permissible combination, column representing responses or objectives and page for each probability level. If it was a deterministic evaluation, the output would be a two dimensional matrix with rows for technology combinations and columns for responses.

8.5.3 Extract True Pareto Front or Layers

In this step, the solutions that are part of the *true* Pareto front or layers are extracted from the data available from the previous step. Here, the Pareto solutions are identified based on the concept of non-domination described in Chapter 4 and 5.

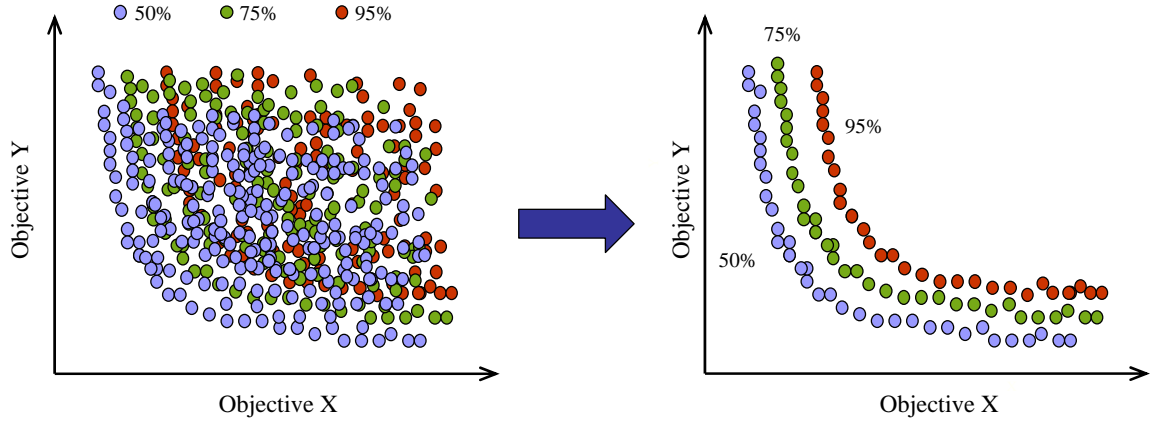


Figure 66: Extracting Pareto Layers in the Objective Space

A non-dominated sorting algorithm based on fitness calculations in SPEA2 is used for this purpose. Here, only raw fitness is required and the points with zero raw fitness are the Pareto optimal points. These points are retained and others discarded. In case of probabilistic results, all technology combinations having a non-dominated point in at least one of the Pareto layers are retained. This step is depicted in Figure 66 where three Pareto layers are extracted in a notional two dimensional objective space. In this figure, a technology combination is represented by three points, each corresponding to a 50%, 75%, and 90% probability level. The non-dominated sorting algorithm extracts the Pareto front in each probability level. And the final result of this step is a union of the technology combinations of all the Pareto layers.

8.5.4 Reduce Dimensionality?

At this stage, the decision on reducing the dimensionality of the objective space is required. The number of objective responses considered in such problems can be large. This set may contain responses that are in fact design constraints but are considered to be objectives because the constraint limits are not well defined. In such situations, the possibility of dimensionality reduction has to be considered. Whenever there are five or more objectives to be optimized, dimensionality reduction step should be

implemented. This is because with increasing dimensions of the objective space, the number of Pareto optimal solutions increase and it becomes difficult for the decision maker to visualize and select appropriate solutions. Moreover, there is a higher chance of two or more objectives being dependent on each other with increasing number of objectives. Thus, the dimensionality of the Pareto hyper-surface may be smaller than that of the objective space. In such a situation, it is advantageous to reduce the dimensionality of the objective space and concentrate on finding the Pareto hyper-surface in its *actual* dimensions.

When it is decided that the dimensionality reduction step is not necessary, the Pareto optimal solutions extracted in the previous step are transferred to the visualization and decision making environment. If, on the other hand, the dimensionality of the objective space has to be reduced, the following step is implemented.

8.5.5 Reduce Dimensions with k -EMOSS Approach

The main aim of this step is to select a subset of objectives to be considered while making technology decisions. As described in Chapter 4, there are two primary techniques of implementing this in the context of Pareto search. One is based on Principle Component Analysis (PCA) as suggested by Deb and Saxena [64]. This method aims at retaining the objectives that can explain most of the variance in the data. The main drawback of this technique is that it does not offer any means of assessing and comparing non-dominated points obtained before and after the dimensionality reduction. Moreover, it selects objectives based on the linear correlations and other types of relations between objectives are not captured. The other technique is based on preserving the dominance structure of the Pareto front as proposed by Brockhoff and Zitzler [65]. The advantage of this technique is that it reduces the dimensions while maintaining the dominance structure of the Pareto points. Thus maximum number

of points are retained in the subset of the Pareto front. The inventors of this technique have devised a measure δ that quantifies the change in dominance structure due to reduced dimensions. Based on this concept of δ , they introduce a problem of minimum objective subset of size k with minimum error (k -EMOSS).

This technique is implemented in the POST methodology using a greedy algorithm for solving the k -EMOSS problem. The algorithm takes the n -dimensional Pareto optimal data set, and a value $k \leq n$ as input and provides the objective subset of size k with minimum error δ , n being the number of objectives. In case of probabilistic Pareto layers, the analysis on only one Pareto layer is sufficient for the purpose. This is because they are globally almost parallel to each other. The k -EMOSS algorithm is implemented iteratively for the objective subset of size $k = 2\text{ton}$. k and the corresponding δ values from the greedy algorithm are plotted and the corresponding objective subsets are noted. Decisions regarding which subset to use are made from this plot and will be discussed in detail in the next chapter.

This is a purely mathematical approach and does not account for any engineering considerations. The designer has to use his or her engineering judgement while accepting its results. It may happen that the technique would deem an objective unimportant, while, from the engineering standpoint, it might be indispensable for the decision making process. The designer would have to include such objectives in the subset even though they were not selected by the algorithm.

8.5.6 Extract Pareto Front or Layers for Selected Objectives

This step is implemented to extract the true Pareto front or layers for the objective subset selected in the previous step. Here, non-dominated sorting algorithm is implemented with the true Pareto set (P) for all objectives as input. The output is a true Pareto set (P') for the selected objectives. This new Pareto set is always going to be a subset of the true Pareto set for all objectives ($P' \subseteq P$). P' can be equal to P if there

is no change in the dominance relationship of the Pareto set due to dimensionality reduction. That is, $P' = P$ if and only if the dropped objectives are truly redundant and their absence does not change the Pareto front. In case of probabilistic Pareto layers, same approach is implemented as in Section 8.5.3. The Pareto points thus extracted are exported to a visualization environment for combinatorial technology exploration and selection .

8.6 *Pareto Optimization*

If, in Section 8.4 it is estimated that the problem complexity is high and all the permissible combinations cannot be evaluated and true Pareto layers determined with available computational resources, Pareto optimization using a stochastic algorithm is implemented. The following subsections describe various steps required for this purpose.

8.6.1 Reduce Dimensions?

The designers decide if the dimensions of the objective space are to be reduced. Whenever there are five or more objectives to be optimized, dimensionality reduction should be investigated. There are a couple of benefits of reducing the number of dimensions when they are more than five. First, it becomes easier to visualize and explore the objective space when it has three or four dimensions. When this number increases, the visualization becomes tougher. Secondly, the stochastic algorithms used for Pareto optimization are more efficient in lower dimensional objective space than they are in the higher dimensional space. Moreover, the Pareto surface in a lower dimensional space can be approximated with fewer points. As observed in one of the previous chapters, for more than 15 dimensional objective space, almost all the solution points would belong to the Pareto hyper-surface.

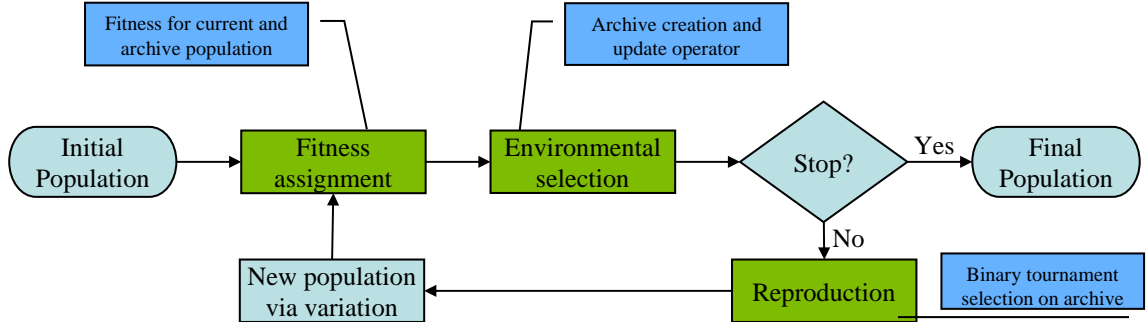


Figure 67: SPEA-II for Deterministic Pareto Optimization

8.6.2 Deterministic Pareto-Optimization

This step is executed when it is required to reduce the dimensionality of the objective space. For this purpose, an approximate Pareto surface in all the dimensions is required. Here, only one layer of Pareto surface is required in case of probabilistic technology impacts. Hence, to reduce the computational cost, deterministic Pareto optimization is implemented using the mean values of technology impacts. The Pareto surface approximation is obtained using the modified Strength Pareto Evolutionary Algorithm II as described in Chapter 5 and its outline is also illustrated in Figure 67.

The initial population of technology combinations is generated randomly. In the next step of *Fitness assignment*, each combination is evaluated by the system model (or surrogate model) and the fitness value is assigned to each combination based on its non-domination characteristic in the objective space. The spacing between points is considered for fitness assignment so as to obtain an even distribution of points on the Pareto front. The constraints are also considered in this step; the technology combinations violating any of the constraints are assigned very high fitness values. The next step is of environmental selection where the best points with lowest fitness values are archived for the next generation. If the stopping criteria of maximum number of generations is not reached, next iteration starts with a *Reproduction* operator. Here, the technology combinations constituting the population for next generation

are selected via a binary tournament on the archived points. Then the new population is generated via *Variation* operators (crossover and mutation), followed by the fitness assignment for this new population. When the algorithm iterates through the maximum number of generations, the archive population at that stage is presented as the final results. This is an approximation of the Pareto front.

8.6.3 Reduce Dimensions with k -EMOSS Approach

With an approximate Pareto front available from the last step, the procedure for dimensionality reduction is the same as described in Section 8.5.5. The only difference is in the data set. In the previous section the true Pareto front was available. While in this step, the dimensionality reduction algorithm uses the approximate Pareto front points to calculate the objective subset. Moreover, there can be some error in selecting the objective subset because of the approximate nature of the Pareto front, but this can be negligible for a large Pareto set.

8.6.4 Deterministic or Probabilistic Pareto-Optimization

After reducing the dimensions of the objective space, deterministic or probabilistic Pareto optimization is carried out on the selected objectives. In case of deterministic technology impacts, Pareto optimization is executed in the manner described in Section 8.6.2. The only difference being the number of objectives considered.

On the other hand, if the technology impacts are uncertain, a probabilistic Pareto optimization is executed using an evolutionary algorithm that accounts for the marginal distribution of technology combinations in objective space. The overall algorithm remains the same as illustrated in Figure 67. The only difference is in the way the fitness is calculated for each population member (each technology combination). For this purpose, the objective vector at different probability levels of interest (for eg. 50%, 75% and 90%) is evaluated using Monte Carlo Simulation (MCS). Thus each combination has a representative point in each Pareto layer. The fitness for points in

each layer is evaluated independently, thus each population member will have multiple fitness values (equal to the number of Pareto layers). The constraints are also accounted for at the Pareto layer level in the same way as described in Section 8.6.2. For the purpose of environmental selection and reproduction operators, the sum of fitness values in all Pareto layers for a population member represents its true fitness in the probabilistic objective space. This way, if a technology combination A has response values that are all non-dominated in their respective Pareto layers, and another combination B with non-dominated objective values in only one Pareto layer; the combination A will have lower overall fitness value relative to B .

The final output of this step are the Pareto layers for each specified probability level. These are basically the objective vectors for each Pareto layer corresponding to a technology combination.

8.7 Exploring and Selecting the Technologies

Once the *true* or approximate Pareto layers for the multi-dimensional space of selected objectives are available via complete evaluation or Pareto optimization respectively, the data is transferred to a visualization and exploration environment. This environment is facilitated by JMP® [53] as previously stated. The data flow for this step is illustrated in Figure 68 with a screen shot of JMP®. Here, the Pareto layers are visualized in a variety of plots and graphs. The DMs can navigate the PLs by setting ranges on the objectives and placing constraints. Each layer can be visualized individually by turning others off. It can be overwhelming for DMs when the dimensionality is large. To overcome this to a certain extent, ideas suggested by Das and Dennis [72] can be implemented. They suggest setting up a hierarchical order of preference among objectives in blocks of two or three. For example, $\{R_1, R_4\}$ may be the most important for some DMs and then $\{R_3, R_5, R_6\}$. Now, the Pareto optimal points could be visualized for each block, starting with the most important, and most

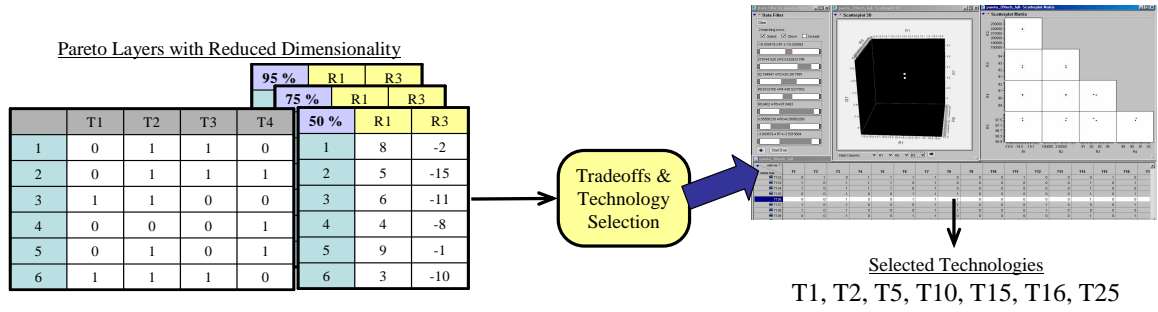


Figure 68: Tradeoffs and Decision Making

appropriate points selected from the block. The DMs can narrow down their selections down the blocks and choose one or more preferred technology combinations. More on this section will be discussed in the following chapter with the help of an example.

The final output of this step and that of POST is a set of promising technology combinations that satisfy the design constraints and DMs preferences. But more than the selected combination, POST provides the DMs and designers the knowledge and understanding of the limits of the design envelope expanded by the technologies available.

8.8 Summary

The purpose of this chapter has been to formulate a technology combinatorial space exploration method that addresses the multi-dimensional nature of the problem and accounts for uncertainties involved with technologies. A method called Pareto Optimization and Selection of Technologies (POST) was formulated by synergistically combining various techniques and methods studied in the previous chapters. Various steps involved in the POST methodology were explained. Once the problem is defined and technologies identified, the complexity of technology combinatorial space is evaluated. If this complexity is low enough for the available computational resources,

a complete enumeration and evaluation of the technology combinations can be executed to search for the *true* Pareto layers. If, on the other hand, the complexity is very high, a stochastic algorithm is suggested to search for an approximation of the Pareto layers. For this purpose, an evolutionary algorithm was designed to search for Pareto front or layers in case of deterministic or probabilistic technology impacts respectively. A dominance based dimensionality reduction procedure was suggested when the dimensionality of the objective space is larger than five. This procedure tries to maintain the relative structure of the Pareto front while trying to identify the redundant objective. Once the Pareto front or layers are identified, they are transferred to a JMP[®] based visualization environment. Here, the technology combinations that are part the Pareto layers are explored for a better understanding of the technology combinatorial space. The limits of new technologies are also identified and finally, the most promising technology combinations can be selected for further study.

CHAPTER IX

EXPLORING TECHNOLOGIES FOR A COMMERCIAL AIRCRAFT

A method called Pareto Optimization and Selection of Technologies (POST) was formulated in the previous chapter to address the primary goal of this research of efficiently exploring technologies for complex systems. This chapter describes the implementation of POST for exploring the technology combinatorial space for a commercial aircraft design problem. Various steps of POST for problem definition and probabilistic Pareto optimization are described in detail. In the later part of this chapter, the discussion is focused towards the systematic exploration of the combinatorial technology space. Different multi-dimensional analysis and visualization techniques are used for this purpose.

9.1 Aircraft Technology Problem Formulation

A predefined technology exploration and selection problem for a large commercial jet aircraft is considered for the implementation of POST method. The original study was undertaken as a part of the NASA GRC research contract [150] at Georgia Tech. The problem was defined under the Vehicle Integration, Strategy and Technology Assessment (VISTA) initiative undertaken at NASA. It involved assessment of 29 technology programs available for a 300-passenger commercial aircraft. Most promising technology combinations are to be identified considering various system level responses.

There are fifteen aircraft responses considered and are listed in Table 16. Out of these responses, the noise (SL, TO, and Ap noise) and emissions (NOx) responses

Table 16: Responses Considered

Responses		
R1	L/D max M0.85 40,000ft (design)	L/D
R2	Empty Weight of Aircraft Without Engines	Empty Wt.
R3	Sideline Noise	SL Noise
R4	Takeoff Noise	TO Noise
R5	Approach Noise	Ap Noise
R6	Cruise Thrust Specific Fuel Consumption	TSFC
R7	Thrust to Weight of Engine	Engine T/W
R8	NOx (Emissions)	NOx
R9	Block Fuel Consumption	BFC
R10	Take-off Gross Weight	TOGW
R11	Direct Operating Cost + Interest	DOC+I
R12	Landing Field Length	LFL
R13	Take-off Field Length	TOFL
R14	Approach Velocity	Ap Velocity
R15	Acquisition Cost	Acq. Cost

are considered as constraints for the purpose of the POST implementation. The remaining eleven responses have to be optimized simultaneously. All the objectives, except L/D and Engine T/W, have to be minimized.

9.1.1 Baseline

A state-of-the-art baseline concept is preferred for the assessment of technologies for a given system. Boeing-777 is considered as the state-of-the-art for a 300-passenger long range commercial aircraft segment. Thus, the baseline for the VISTA study was a 777-like aircraft on which various technologies are evaluated. The response values for the baseline are calculated (with all technologies inactive) and listed in Table 17. This table also defines the four inequality constraints used for the current implementation. The constraints are defined by fixing more than 2% reduction in the baseline noise values and around 15% reduction in the NOx values. These constraints and the corresponding reduction in the objective space represent the combinatorial technology space that would be of interest to the designers and decision makers.

The researchers from Georgia Tech and VISTA team identified sixty system level

Table 17: Responses and Constraint Values for Baseline Aircraft

Response	Baseline	Constraint
L/Dmax M.85 40,000 ft (design)	19.22	
Empty Weight of Aircraft without Engines, lbs	234840	
Sideline Noise, EPNdB	94.93	≤ 93
Take-off Noise, EPNdB	91.98	≤ 90
Approach Noise, EPNdB	98.32	≤ 96
Cruise TSFC, lb/lbf.h	0.56	
Thrust to Weight of Engine	3.91	
NOx (Emissions), kg/LTO	52.01	≤ 44.2
Block Fuel Consumption, lbs	85294	
Take-off Gross Weight, lbs	659020	
Direct Operating Cost + Interest, cents/ASM	4.35	
Landing Field Length, ft	5828.40	
Take-off Field Length, ft	9532	
Approach Velocity, kts	122.70	
Acquisition Cost, million \$	119.72	

design variables or technology k -factors for this problem [150]. The simulation models created for technology evaluation are based on these k -factors; that is, they are the inputs to the simulation models. The values of these variables are fixed for the baseline aircraft and the new technologies are assessed by estimating their relative impact, with respect to the baseline, on various k -factors. The technologies are then evaluated for the aircraft with these models by accounting their impacts on the k -factors.

9.1.2 Modeling and Simulation

To evaluate technologies for the VISTA study, various physics based numerical analysis tools were integrated. These included NASA's Numerical Propulsion System Simulation (NPSS) for engine responses, Weight Analysis of Turbine Engines (WATE) for weights, Flight Optimization System (FLOPS) for aircraft responses, Aircraft Noise Prediction Program (ANOPP) for noise, and Aircraft Life Cycle Cost Analysis (AL-CCA) for economic responses [150]. The complexities with this integrated simulation environment result in long execution time and it becomes prohibitive to evaluate large number of technology combinations in a reasonable time. To address this issue,

surrogate models were created.

These surrogate models, created for each aircraft response by the researchers at Georgia Tech, are used for evaluating different technologies. These surrogate models are in the form of Response Surface Equations (RSEs) that were created using the Response Surface Methodology (RSM) as described by Myers and Montgomery [151], and Box and Draper [152] among others. RSEs are typically second order polynomial equations as given by Equation 40.

$$R = \beta_0 + \sum_{i=1}^k \beta_i x_i + \sum_{i=1}^k \beta_{ii} x_i^2 + \sum_{i=1}^{k-1} \sum_{j=i+1}^k \beta_{ij} x_i x_j + \varepsilon \quad (40)$$

where,

R = response of interest

β_0 = intercept term

β_i = coefficient for first order terms

β_{ii} = coefficient for second order terms

β_{ij} = coefficient for cross-product terms

x_i = main effect of independent variables

x_{ii} = quadratic effect of independent variables

x_{ij} = second order interaction effects of independent variables

ε = associated error

The coefficients of this equation are usually determined by least square analysis of the experimental data. For the required experimental data, Design of Experiments (DoE) is used to create statistically important experiments. Each experimental unit is then evaluated using the complex numerical simulation environment and the required response data is generated. The RSEs thus created are a function of sixty technology metrics or k -factors for the current problem.

9.2 *Technology Data*

The 29 technologies identified for this problem are denoted as T1, T2, ..., T29 and listed in Table 18. The impact of these technologies on the aircraft were elicited from the technology and system experts using the Technology Metrics Assessment and Tracking (TMAT) process [21]. This process was enabled by the Technology Audit Sheets for each technology which were filled out by the respective technology experts. Additional information was gathered from interviews with technologists and other data provided by them. This helped in identification of the k -factors and also in quantifying the technological uncertainty. For the current problem, the impact of these technologies on the k -factors is not deterministic and three values, pessimistic, optimistic, and most likely have been defined that indicate the uncertainty in each impact. Based on these three values, a generalized beta distribution is defined by calculating the location a , scale $(b - a)$, α , and β parameters for each technology impact.

There are some compatibility constraints among 29 technologies. These are represented by the technology graph shown in Figure 69. Most of these constraints are present because the corresponding technologies are competing with each other. The incompatibility between T1 and T2 is present because the slotted wing in T1 would effect the design of the bump in T2. Technologies T1 and T16 are incompatible because not much is known about their combined effects. The interactions among T8, T12, T13, and T14 are more complicated. From this set of four technologies, any combination of one, two, or three can be implemented, all four cannot be included simultaneously. Moreover, they have special implementation scheme based on the number of technologies considered; this is take into account while evaluating the RSEs. The remaining technologies are independent of one another and can be implemented in any combination.

Table 18: Technologies Considered

Tech. No.	Technology Description
T1	High Speed Slotted Wing
T2	Transonic Adaptive Bump
T3	Sensory Materials and Damage Science
T4	ST Manufacturable Large Structures
T5	Slat-Cover Filler
T6	Landing-Gear Noise Reduction
T7	Core Cowl Acoustic Liner
T8	Installation Improved Chevron Nozzles
T9	Flap Trailing Edge Treatment for Jet Interaction
T10	Soft Vanes (Stators)
T11	Fan Duct Acoustic Splitter
T12	Offset Stream Technologies
T13	Chevron Vortex Stabilization
T14	Fluidic Chevrons
T15	Inlet Blowing/Liner Integration
T16	Herschel-Quincke (HQ) Tube/Liner Integration
T17	Low Nox Combustor Development Type A
T18	Low Nox Combustor Development Type B
T19	3000F Ceramic Matrix Composite (CMC) combustor materials
T20	3000F metallic combustor materials
T21	2 Stage Proof of Concept Compressor
T22	Highly loaded High Pressure Turbine (HPT)
T23	Highly loaded Low Pressure Turbine (LPT) with aggressive duct
T24	Fan Containment
T25	Nickel Disks
T26	Lightweight Single Crystal Blade Alloy
T27	Low Conductivity Thermal Barrier Coating (TBC)
T28	2700 deg Ceramic Matrix Composite (CMC) Liner
T29	2700 deg Ceramic Matrix Composite (CMC) Vane

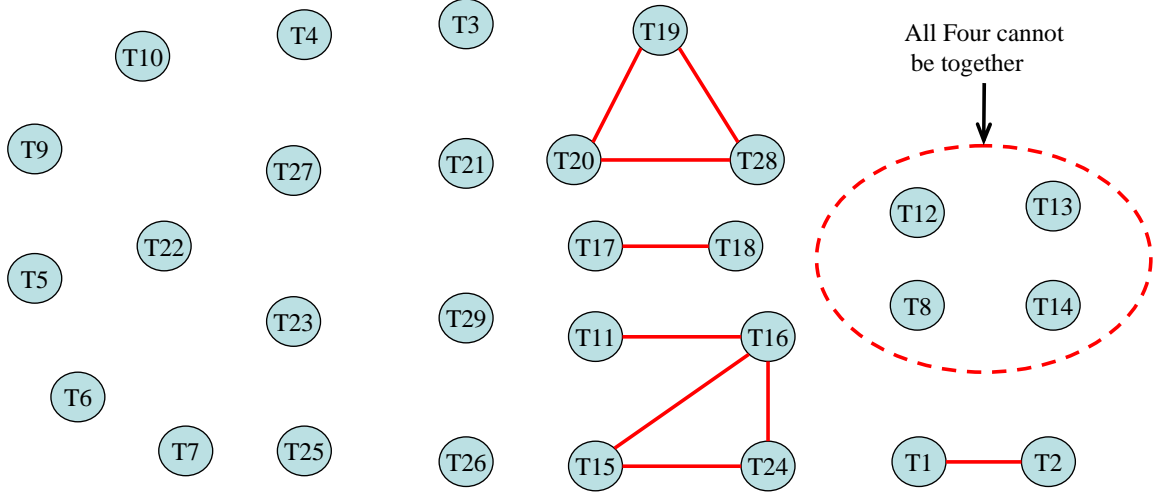


Figure 69: Technology Graph

9.3 Complexity of Technology Graph

Based on the technology graph from Figure 69, the average complexity of the problem is calculated in this step using Equation 41. Here, $t = 29$ and the value for probability of edge is obtained by dividing the total number of edges present in the graph by the total number of possible edges. As observed from the technology graph, there are 9 edges among 11 distinct technologies. In case of the set with four technologies, at the most any three can be included simultaneously. To account for this relationship, one *virtual* edge can be considered among them. Thus, for the purpose of calculating the average number of independent sets, there are 10 edges in the graph. Based on 10 edges and 29 technologies, $p = \frac{10}{\binom{29}{2}} = 0.0246$. Thus, from Equation 41, the average number of permissible combinations is $I_{G^p} = 62,571,000$. This is out of the total combinations $2^{29} = 536,870,912$. Even though the number of permissible combinations is considerably less than that of all combinations, it is still significantly large. Available computational resources (software and hardware) in the form of MATLAB operating on a Windows based desktop computer cannot handle about 62 million combinations to filter out the *true* Pareto frontier. Moreover, probabilistic

evaluation of these many combinations is not possible in a reasonable amount of time. Thus Pareto optimization route is adopted for this problem.

$$I_{G^p} = \sum_{m=0}^t \binom{t}{m} (1-p)^{m(m-1)/2} \quad (41)$$

9.4 *Reduce Dimensionality?*

This step is to decide if dimensionality reduction is desired for the problem. Out of fifteen responses, there are eleven objectives to be minimized. The possibility of dimensionality reduction has to be considered whenever there are five or more objectives to be optimized. The current example has eleven objectives, out of which there are a few that may be dependent on one another. For example, Empty Wt. and TOGW are correlated; same can be said about TSFC and Block fuel consumption. Considering the presence of such correlations among the objectives, it is prudent to reduce the number of objectives considered for optimization. Moreover, the presence of redundant objectives tend to reduce the efficiency of Pareto optimization algorithm. Thus, it is decided to implement the dimensionality reduction procedure.

9.5 *Deterministic Pareto Optimization*

A representative Pareto set in all eleven dimensions and bounded by the four constraints is required for investigating the dominance structure for dimensionality reduction. In this step, a Pareto hyper-surface is obtained with Pareto optimization considering deterministic impacts of the technologies. Towards this end, SPEA2 with constraints listed in Table 17 is implemented on 29 technologies starting with a random population of 100 technology combinations. The *most likely* values of technology *k*-factors are used for evaluating technology impacts on the aircraft. The simulation is run through 300 generations with an archive size of 1500 points. Total time required for this simulation is around two hours. The technology-presence on the resultant

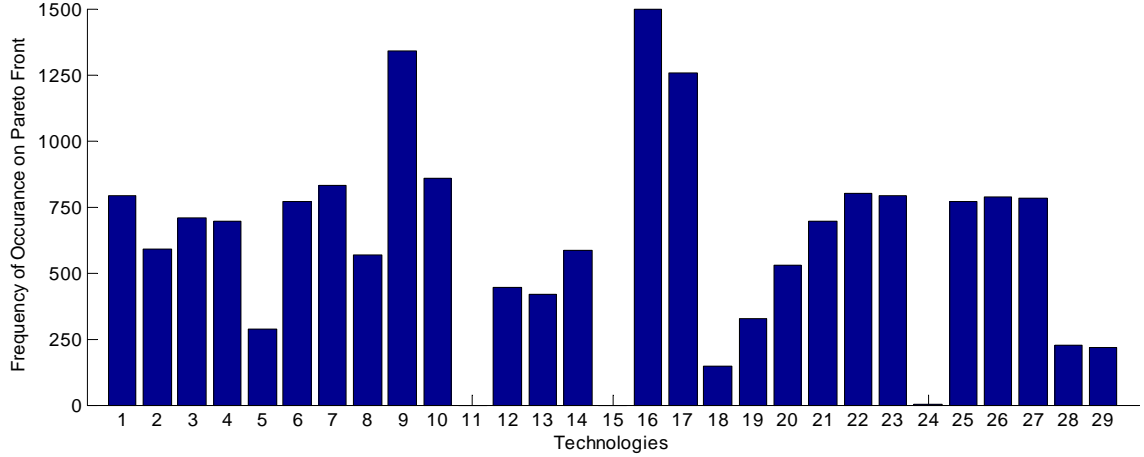


Figure 70: Technologies Present on 11 Dimensional Pareto Front

Pareto front can be observed from Figure 70. This figure shows the frequency of occurrence for each technology on the deterministic Pareto front in eleven dimensional objective space. It can be observed that technology T9, T16, and T17 are present on most of the points on Pareto front; on the other hand, T11, T15, and T24 are absent throughout the entire Pareto front.

The Pareto front is in the form of a data matrix of 1500 rows and 11 columns; each row represents a point on the 11 dimensional Pareto hyper-surface. This data is used in the next step of dimensionality reduction.

9.6 Dimensionality Reduction

In this step, dimensionality reduction procedure is implemented on the Pareto front data obtained from the previous step. A k -EMOSS analysis, as proposed by Brockhoff and Zitzler [70] and explained in Section 4.3.1.2 of Chapter 4, is implemented on 1500 archive points to investigate the prospects of dimensionality reduction. One important aspect that should not be overlooked for this analysis is the scale of various objectives. The measure for empty weight (Empty Wt.) and take off gross weight (TOGW) is in a hundred thousands range and that for thrust specific fuel consumption (TSFC)

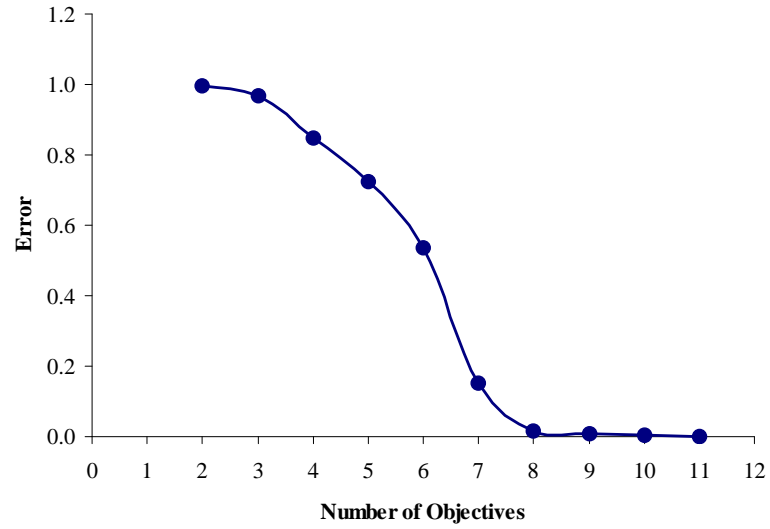
is less than one. The k -EMOSS algorithm searches for an objective subset of size k which has minimum dominance error δ . This error is calculated in absolute terms and not relative to the scale of the objectives; thus, the results will be skewed when some of the objectives have large measures than others. To address this problem, all the objective values for 1500 points are normalized between 0 and 1. Now, all objectives have same measure while their relative position on the Pareto front is preserved.

A greedy algorithm by Brockhoff [153] for k -EMOSS analysis on the the normalized data is implemented with k values ranging from 10 through 2. Here, the algorithm is executed repeatedly for each value of k and the resultant objective subset of cardinality k and having minimum dominance error δ is recorded. The results obtained are listed in Table 19. This data is also plotted in Figure 71 with cardinality of objective subset k on the horizontal axis with corresponding δ values on the vertical. It can be observed from the figure that when $k = 11$, that is all objectives are considered, the dominance error is naturally zero. Now, moving left on the horizontal axis, the dominance error increases with each reduction in the k value. From $k = 10$ through $k = 8$, there is only minor increase in δ and the dominance structure is almost similar to the Pareto front of 11 dimensions. There is some increase in δ value going from $k = 8$ to $k = 7$ but the biggest jump in dominance error occurs from $k = 7$ to $k = 6$. This jump in δ indicates the importance of having the objective *Cruise TSFC* (R6) that is present in the subset with $k = 7$ and absent in the one with $k = 6$. The subset with cardinality 7 is of interest for Pareto optimization. This objective set includes { L/D, Empty Wt., TSFC, Engine T/W, DOC+I, LFL, TOFL }. One important objective which is an important part of the tradeoff exercise, and absent from the current subset, is the *Acquisition Cost* (R15). This is also considered for Pareto optimization; and now the final objective set for Pareto optimization is: { L/D, Empty Wt., TSFC, Engine T/W, DOC+I, LFL, TOFL, Acq. Cost }.

This reduced set of objectives is obtained with the help of mathematical analysis.

Table 19: Dimensionality Reduction With k -EMOSS

k	Objective Subset	Error
11	1,2,6,7,9,10,11,12,13,14,15	0.000
10	1,2,6,7,9,10,11,12,13,14	0.003
9	1,2,6,7,10,11,12,13,14	0.006
8	1,2,6,7,11,12,13,14	0.017
7	1,2,6,7,11,12,13	0.152
6	1,2,7,11,12,13	0.536
5	1,7,11,12,13	0.726
4	1,7,11,12	0.849
3	1,7,12	0.968
2	1,7	0.996

**Figure 71:** k -EMOSS Analysis for 11 Objectives

In most instances, this will correspond to engineering judgement. For example, *Cruise TSFC* and *Block Fuel Consumption* are usually correlated and TSFC is the preferred metric among the two for engineering decision making. With current dimensionality reduction procedure, *TSFC* is selected and *BFC* is left out; this corresponds to the engineering judgement. On the other hand, *Empty Wt.* and *TOGW* are also correlated and if given a choice, *TOGW* would be preferred over *Empty Wt.* But, with the current dimensionality reduction procedure *Empty Wt.* is chosen over *TOGW*. This is because the dimensionality reduction technique is based on preserving the dominance structure, that is it tries to maintain the structure of the Pareto front while reducing the dimensions. If the *Empty Wt.* is replaced by *TOGW* in the objective subset with cardinality $k = 7$, the corresponding dominance error $\delta = 0.692$. This is significantly higher than $\delta = 0.152$ for the original subset. Thus, even though including *TOGW* in place of *Empty Wt.* would be considered a sound engineering judgement, it would degrade the overall tradeoff potential of the Pareto solutions objective space.

9.7 Probabilistic Pareto Optimization

Because of the uncertainties present in the technology impacts, probabilistic analysis using Monte Carlo Simulations (MCS) on the surrogate system model represented by the RSEs is implemented. Three probability levels, 50%, 75%, and 95%, for each objective value are calculated for the technology combinations using the empirical marginal CDFs obtained via MCS. The random samples of the k -factor values for MCS are created from the generalized beta distributions defined earlier. The sample size is fixed at 500. According to Equation 42 as discussed in Chapter 6, this sample size corresponds to an error of 8.9%, 5.1%, and 2.0% for p -levels of 50%, 75%, and 95% respectively. The p -level values calculated in this step are used for Pareto optimization.

$$\varepsilon = 2\sqrt{\frac{(1-p)}{np}} \quad (42)$$

The probabilistic Pareto optimization is implemented with eight objectives selected in Section 9.6. The evolutionary algorithm is continued through 300 generations with a population of 100 technology combinations and an archive size of 1500. The mutation and crossover probabilities are fixed at 5% and 80% respectively. The gene correction algorithm as described previously is implemented, to address incompatibilities and other technological constraints, just before the evaluation of population members. The algorithm is designed for objective minimization, hence negative values of *Lift/Drag Max* (R1) and *Thrust to Weight Ratio of Engine* (R7) as they are to be maximized. Responses at three probability levels are evaluated for each members as explained in the previous section. Fitness for each member is calculated based on these three values. The constraints for noise and emission responses are implemented by means of a penalty function. Whenever the response constraints are violated at any probability level, the fitness of that technology combination is increased proportional to the constraint violation. This is represented by Equation 43.

$$f_i = f_i + c_i \times f_{max} \quad (43)$$

Here, f_i is the fitness of i^{th} individual in the population and f_{max} is the maximum fitness value in that generation. c_i is the number of constraints violated by the i^{th} technology combination. That is, if a combination violates all four constraints, $c_i = 4$ and if it violates only one of the four constraints, then $c_i = 1$. If no constraint is violated by the technology combination, then $c_i = 0$, and the penalty function is not imposed.

After the fitness values are calculated, environmental selection or archiving is implemented. Reproduction, crossover, and mutation operators follow next and a new population for the following generation is created. At the end of 300 generations, the archive with its rows of technology combinations and corresponding response values for all three p -levels is obtained. These 1500 archive points form the Pareto layers in an 8-dimensional objective space. Time required for this simulation was around 36

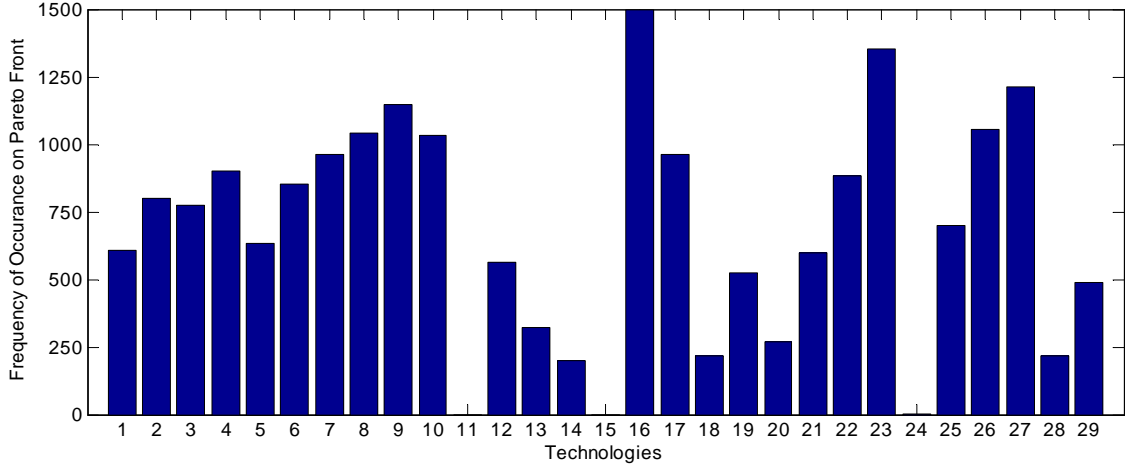


Figure 72: Technologies on 8 Dimensional Probabilistic Pareto Layers

hr 30 min on a 4 processor 2 Ghz Xeon dual core machine with 8 GB of RAM.

A rough estimate of the importance of various technologies can be gained at this stage by a technology bar graph as seen in Figure 72. This chart shows the presence of each technology on the probabilistic Pareto layers. It can be observed that almost all technologies, except T11, T15, and T24 have considerable presence on the Pareto layers. Some significant technologies can also be identified. For example, T16, T23, T27, T9, and T26 are present in more than 800 combinations out of 1500. Moreover, T16 is present in all 1500 archive points (technology combinations) and cannot be ignored in the final solution. These can be considered as very active on the Pareto layers and require due attention in the next step of technology selection.

The result of this step is a set of archive points. This represents the three probabilistic Pareto layers in 8-dimensions. Moreover, in this application, there are four environmental constraints considered. Thus, the Pareto layers obtained are composed of only the points that meet the noise and emission constraints.

9.8 Exploring and Selecting the Technologies

The 1500 technology combinations, each with three p -level values each for eight objectives and four constraints are exported to the visualization and analysis environment enabled by JMP® [53]. This is illustrated in a screen-shot of JMP® in Figure 73. As shown in the figure, each row of the data matrix represents a technology combination. The technology columns denote the presence or absence of a particular technology in the combination by a binary value of 1 or 0 respectively. Adjacent to the technology columns are the columns for objective and constraint responses. The constraint responses are also included in this data set to provide the flexibility for tightening the constraints and selecting a technology combination based on that.

As there are three probability levels considered, there are three Pareto layers. Thus, each technology combination results in a set of three response values. To accommodate this three dimensional data structure in JMP, each technology combination is represented in three rows, one each for a different p -level responses. Thus, whereas the number of technology combinations in the archive of probabilistic Pareto optimization is 1500, the number of rows required for this data is archive size \times number of Pareto layers = $1500 \times 3 = 4500$. To differentiate each Pareto layer, three columns of *Pareto layer indicator* are added to the data. Here, the column has value 1 if the row belongs to that particular Pareto layer. This is illustrated in Figure 74.

The last five columns are indicator columns for lower dimensional Pareto surfaces. When the objective space is of high dimensionality, it is desired to investigate the lower dimensional Pareto fronts that are the subsets of the higher dimensional surfaces. For this purpose, four 2-D and one 3-D Pareto layers are extracted from the 8-D Pareto layers. The points lying on the sub-dimensional Pareto fronts are indicated by 1 in the respective column.

Once the data is configured in this manner, rows can be selected and data filtered based on the Pareto layer of interest (objectives and p -level).

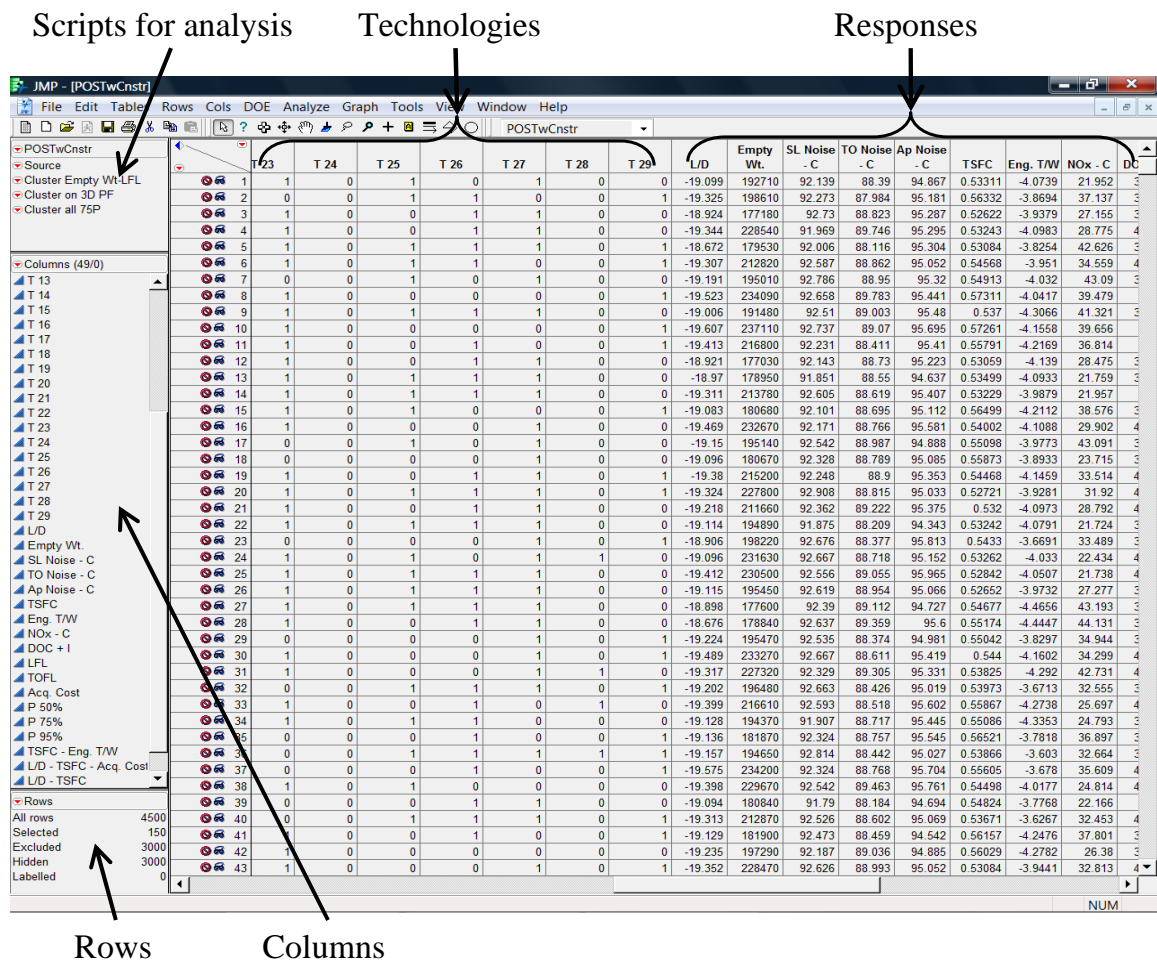


Figure 73: Pareto Layer Data in JMP®

Pareto layer indicator

Sub-dimensional Pareto front indicator

Window Help											
POSTwCnstr											
	TOFL	Acq. Cost	P 50%	P 75%	P 95%	TSFC - Eng. T/W	L/D - TSFC -...	L/D - TSFC	L/D - Acq. Cost	TSFC - Acq. Cost	
7	9336.6	103.77	1	0	0	0	0	0	0	0	
2	9895.7	106.72	1	0	0	0	1	0	0	0	
5	9472	98.604	1	0	0	0	1	0	0	1	
7	9260	114.89	1	0	0	0	1	0	0	0	
2	10037	99.842	1	0	0	0	0	0	0	0	
7	9103.8	111.33	1	0	0	0	0	0	0	0	
6	9654.5	104.25	1	0	0	0	0	0	0	0	
6	9107	118.44	1	0	0	0	0	0	0	0	
2	9058	101.61	1	0	0	0	0	0	0	0	
5	9804	118.49	1	0	0	0	0	0	0	0	
3	9886.8	110.65	1	0	0	0	0	0	0	0	
9	9463.8	96.972	1	0	0	1	1	0	0	1	
2	10143	98.639	1	0	0	0	0	0	0	0	
9	9892.6	111.36	1	0	0	0	0	0	0	0	
7	9289.9	98.286	1	0	0	0	0	0	0	0	
7	9909.1	116.53	1	0	0	0	0	0	0	0	
4	8990.8	105.37	1	0	0	0	0	0	0	0	
6	9352.8	100.86	1	0	0	0	0	0	0	0	
9	9876.7	110.27	1	0	0	0	1	0	0	0	
7	9086.8	116.06	1	0	0	0	0	0	0	0	

Figure 74: Identifying Pareto Layers and Sub-dimensional Pareto Fronts

9.8.1 Scatter Plots

The first plot through which the multi-dimensional Pareto layers should be visualized is the scatter plot matrix also known as the draftsman's plot. This plot for the 8-dimensional Pareto layers is illustrated in Figure 75. This plot enables one to see the entire 8-dimensional objective space in a single glance. The limits of technologies are clearly visible in this plot. For example, given the environmental constraints, the minimum acquisition cost that can be achieved with given technologies is around \$ 96 million at 50% probability level. If even lower acquisition cost is the aim, more cost-reducing technologies or other avenues have to be investigated. The trends in the objective space are noticed in this plot. For example, the plot of Acq. Cost and Empty Wt. is almost a diagonal line indicating that these two objectives are correlated. This is true for commercial aircrafts and cost-weight relationships are a well known tool used by designers to estimate cost of an aircraft. Thus scatter plot matrix can help validate to a certain extent the models and assumptions used in evaluating the technologies. This plot also helps identify the objectives that have the maximum potential of making tradeoffs. It can be seen that objective pairs like Eng. T/W-TSFC, Acq. Cost-L/D, etc. have high potential for tradeoffs. On the other hand not much tradeoff is involved in selecting technology combinations for Empty Wt.-Acq. cost.

After investigating the scatter plot matrix, it is of general interest to focus on specific sets of two or three objectives where the possibility of tradeoff is high. This is also aligned to the ideas suggested by Das and Dennis [72] of visualizing multi-dimensional Pareto fronts in blocks of two or three dimensions. The sub-dimensional Pareto front indicators are very useful for this purpose. As an example, Figure 76 illustrates the filtering of Eng. T/W-TSFC Pareto layers from the 8-D Pareto data set. The symbols and colors used for the points are the same as in Figure 75. JMP® provides a data filter tool shown in the figure where a range of column values can

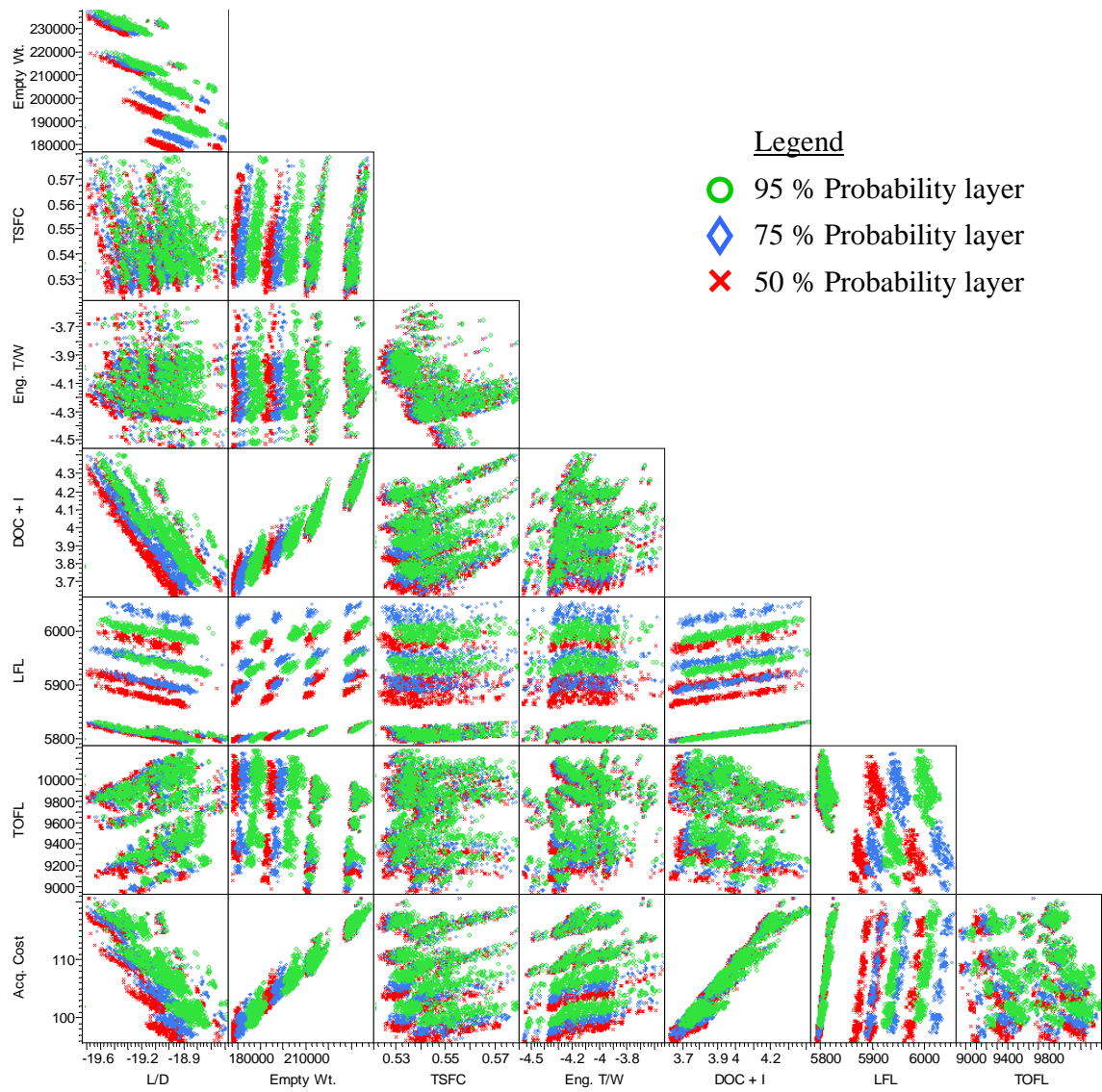


Figure 75: Scatter Plot Matrix for 8-Dimensional Pareto Layers

be fixed and the software selects and shows the rows that are within those bounds. For the current application, the indicator column for TSFC–Eng. T/W is set to 1. This selects the technology combinations that are on the Pareto layers of these two objectives. There are 79 rows out of 4500 that are selected. It should be noted that the cardinality of each Pareto layer may not be the same in this example. A particular technology combination may be present on the Pareto layer of 95% p -level but not on 50% Pareto layer. This happens because the point corresponding to 50% p -level may not be Pareto optimal in the given dimensions.

It is also of interest to study the *drift* or movement of technology combinations between the Pareto layers with different p -levels. Working with the previous example of TSFC–Eng. T/W Pareto layers, Figure 77 illustrates the movement of a technology combination between three Pareto layers in two dimensions. This can also be visualized in multiple dimensions with the help of the scatter plot matrix. The selected technology combination is present on all three 2-D Pareto layers which may not be the case for every combinations as explained earlier. It can be observed that the three points on three different layers need not be connected by a straight line. This can be attributed to the fact that the marginal distributions on different objectives for a given technology combinations may not be similar in their shape. That is to say, the joint probability distribution of a technology combination is not always symmetric over all the axis.

The three dimensional Pareto layers can be visualized using a dynamic 3-D scatter plot. Such a plot for three objectives of L/D–TSFC–Acq. cost is illustrated in Figure 78. The points are selected using the data filter on the indicator column for these Pareto layers. The plot can be rotated and spined to view the layers from any angle. The points of interest can be selected in this plot and can be also seen in multi-dimensional scatter plot matrix. It is interesting to note that there are 1419 points in these 3-D Pareto layers as compared to only 79 in the previous 2-D example.

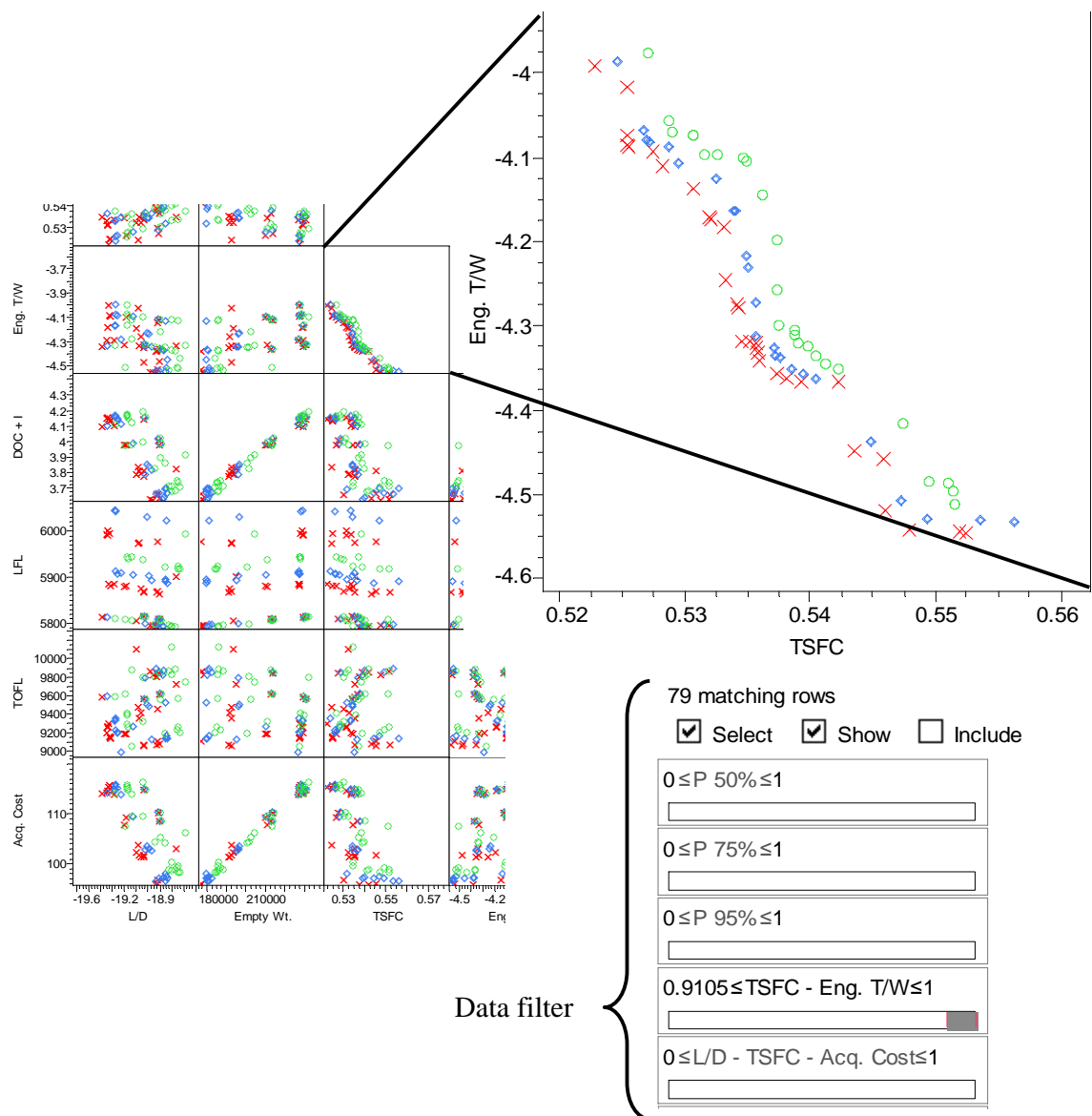


Figure 76: Visualizing Sub-Dimensional Pareto Layers

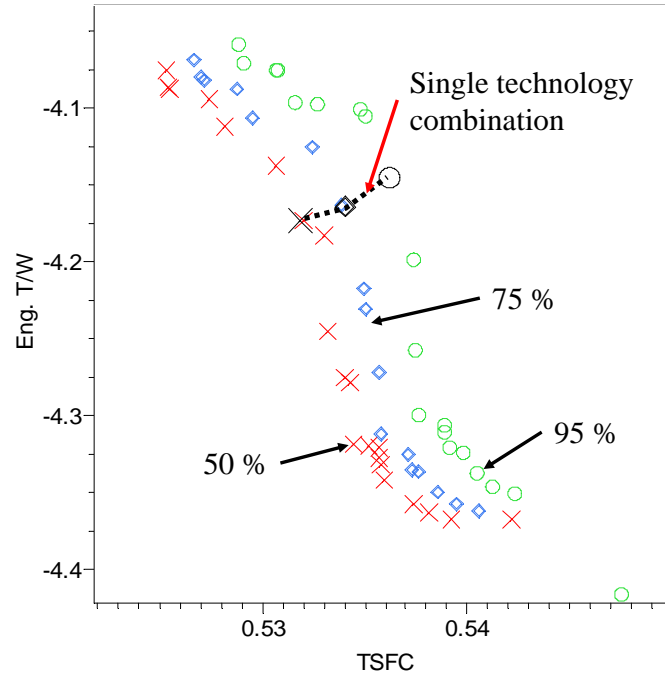


Figure 77: Drift of a Technology Combination Over Pareto Layers

This illustrates the considerable increase in the size of Pareto fronts with increasing dimensions. For a two dimensional Pareto front, it is only an edge, while in a 3-D case the front is a surface and hence more points.

9.8.2 Clustering

It can be observed from the scatter plot matrix in Figure 75 that most of the solutions tend to group or cluster together in the objective space. This happens because of the combinatorial nature of the technology selection space. When certain technologies are present in combinations, they tend to cluster together; but when they are switched off or certain other technologies are added to the combinations, there is a noticeable shift in their positions in the objective space. To study this behavior, clustering analysis is employed in JMP®.

Clustering is a multi-variate analysis technique of grouping together data with similar values. Thus points from one cluster are more similar to each other than

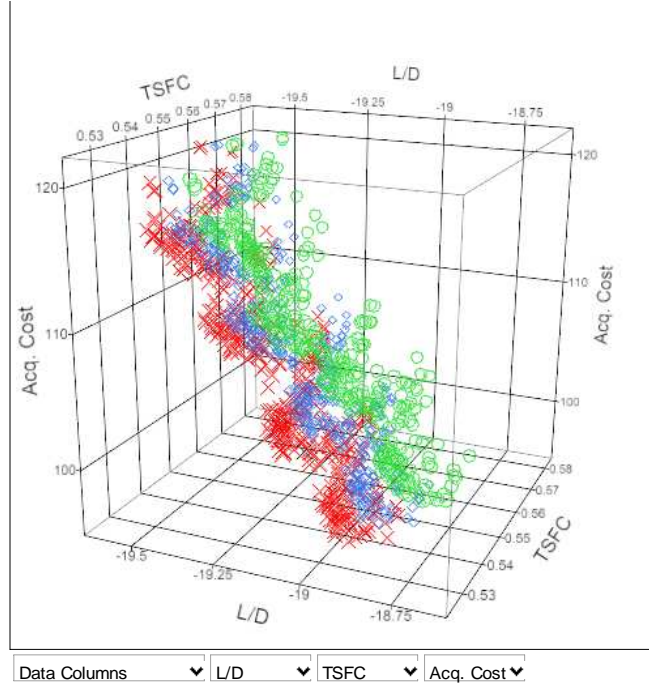


Figure 78: Three Dimensional Pareto Layers

the points from different clusters. This technique helps compartmentalize the multi-dimensional objective space so that one can focus their search in a smaller section of the objective space. The technique used for clustering in this application is called Hierarchical clustering. [53] This is an iterative process starting with each point as its own cluster. At each step the algorithm calculates the distance between each cluster and combine the closest one. This process is depicted in a *dendrogram*. To illustrate the cluster analysis, an example case with an 8-D Pareto layer of 75% is considered. Clustering is implemented in two dimensions of Empty Wt. and LFL. The dendrogram for this analysis is illustrated in Figure 79. Once the analysis is completed and a dendrogram created, the data can be divided into any number of clusters between the number of data points considered to 1. As shown in Figure 80, there are 16 distinct clusters visible in two dimensions. These clusters are sorted accurately by the hierarchical cluster analysis when the number of cluster is fixed to 16.

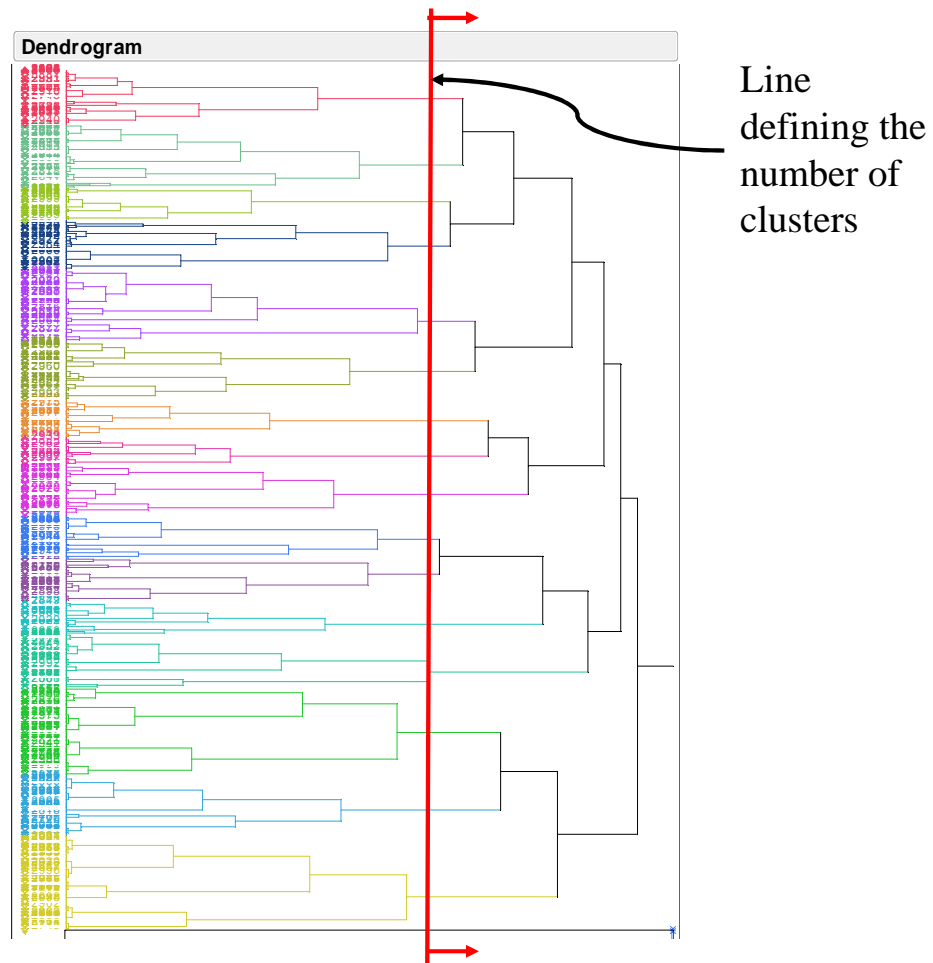


Figure 79: Dendrogram for Cluster Analysis in Two Dimensions

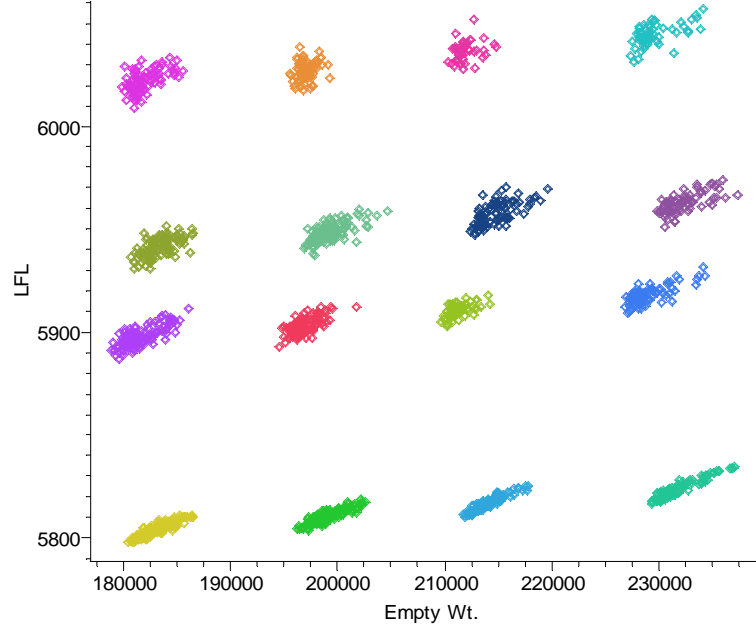


Figure 80: Clusters in Two Dimensions

A two dimensional example is used here for the ease of understanding but this process can be used for any number of dimensions. Any of the clusters can be selected from the dendrogram and studied for prominent technologies. For example, cluster number 16, which is at the lower left corner of Figure 80 has 165 technology combinations. In this group, T3, T4, and T16 are present in every 165 combinations. On the other hand, T1, T5, T11, T15, and T24 are completely absent. Thus, affinities of technologies for a particular region of objective space can be explored.

9.8.3 Strategies for Visual Exploration and Decision Making

The visualization and investigation tools are most effective when the data matrix and plots are viewed simultaneously as illustrated in Figure 81. Thus any technology combination selected on a data matrix can be viewed immediately in a multi dimensional scatter plot. Or, if a point of interest is selected on one of the plots, the corresponding technology combination can be viewed in data matrix. Such a visualization and exploration exercise requires the data to be viewed on a large scale. For example, facilities such as Collaborative Visualization Environment (CoVE) as described by

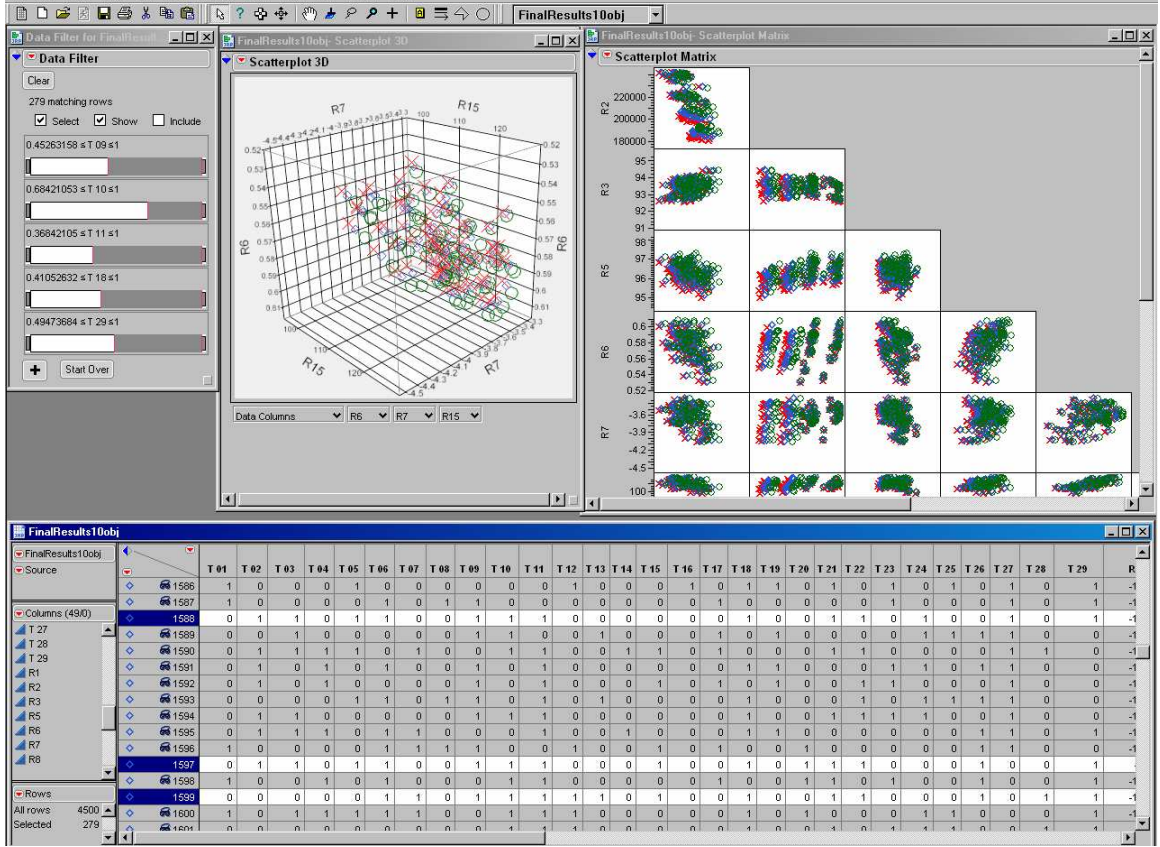


Figure 81: Screen Shot of JMP®

Mavris [154] and Osburg [155] and are ideal for this purpose.

Even when the visualization environment is created, decision making is still not an easy task. Specific strategies for visual exploration and decision making have to be implemented in order to select the most appropriate technology combination. As a first step, it is suggested to select a few blocks of two or three objectives that offer the most possibilities of tradeoffs, and are important to the designers and decision makers. Then extract the Pareto fronts in these sub-dimensions. For the current problem, four two-dimensional and one three-dimensional objective subspaces are identified as important from decision making perspective. These are {Eng. T/W, TSFC }, {Avg. Cost, L/D}, {L/D, TSFC}, {Avg. Cost, TSFC} and {Avg. Cost, L/D, TSFC}. Pareto optimal points for each of the Pareto layers in these subspaces

are identified. The indexes of these points are marked in the indicator columns of JMP[®] data table. These points can be switched on or off as per requirement using the data filter tool on indicator columns. For example, Figure 76 plots the cross section of design space for Thrust/Weight and TSFC showing only the Pareto optimal points in these two dimensions. All other points are turned off. Lets consider that tradeoff in these two dimensions is most important. Points of interest are manually selected from this plot. The position of these points with respect to other axis is checked in the adjacent scatter plot matrix. The process is repeated for other subspaces mentioned. In a large-dimensional objective space as here, it is highly unlikely to find points that exist simultaneously in more than one two-dimensional slice of the Pareto front. Hence one has to be careful should not to select only a few points from the first 2 or 3-D Pareto front.

Another strategy for exploring the combinatorial space is by selecting a particular Pareto layer. If one is interested in high level of confidence, the Pareto layer with 95% confidence can be turned on and data visualized for this layer. With the help of Pareto layers, the tradeoffs can not only be made among the objectives but also in the level of risk the designer is willing to take to achieve performance gains. It can be observed in the previous figures that the Pareto layer with 50% probability level corresponds to a better objective values that the ones with higher probability levels. Thus if the designers and decision makers are interested in better performance and economic values, the decisions can be made based on Pareto layers corresponding to 50% or 75% probability values.

Setting artificial constraints on the objectives is also a good strategy of down selecting the technology combinations. The space of interest can be defined by selecting limits (usually upper limits) on various objectives. This will reduce the combinatorial space considerably facilitating the ease of decision making. Placing some upper limits on objectives for Acq. cost, L/D, and TSFC, the combinatorial space is reduced. For

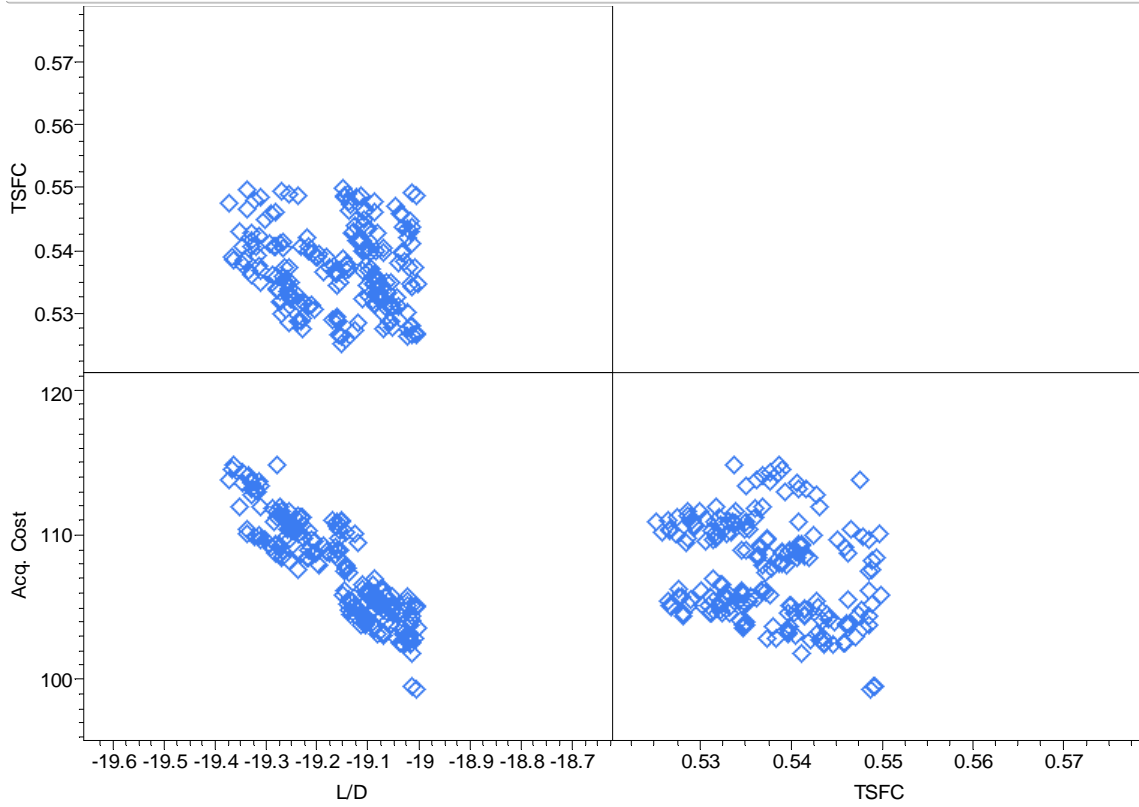


Figure 82: Scatter Plot Matrix with 75% Pareto Layer

the purpose of this exercise, 75% Pareto layer is considered. The limits placed on the objectives are $\text{Acq. Cost} \leq 115$ \$Mil, $L/D \geq 19$, and $\text{TSFC} \leq 0.55$ lb/lbf.h. After placing these limits, there are 206 points remaining on the ten-dimensional Pareto layer from a total of 1500 points. A three dimensional scatter plot matrix for this example is illustrated in Figure 82.

Now, lets consider that the decision makers are interested in a solution that is Pareto optimal on the Eng. T/W–TSFC cross section. For this, the point on the Pareto layer of Eng. T/W–TSFC are selected using the data filter tool. Out of 204, there are only five points that satisfy this criteria. These points are plotted in Figure 83. Out of these five points, two technology combinations with maximum engine thrust to weight ratio are selected as shown in the figure.

These two selected technology combinations are now plotted on a scatter plot

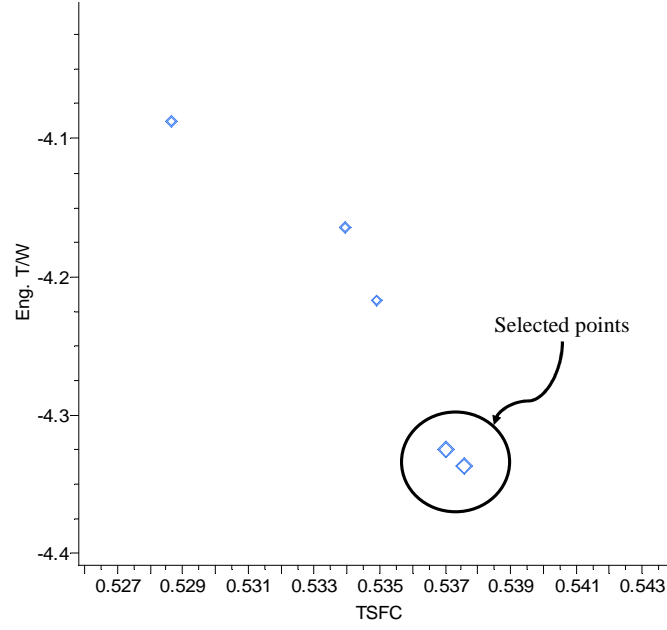


Figure 83: Selected Technology Combinations from 75% Pareto Layer

matrix for all eight dimensions and all three Pareto layers. As observed in Figure 84, one technology combination has considerably more uncertainty associated with itself than the other; especially, in terms of objectives such as DOC+I, Empty Wt., and Acq. cost. This can be noted based on the distance between the three p -level points for a given technology combination. More the distance between 50% and 95% points, higher the uncertainty associated with that combination.

Similar observation can also be made from the parallel co-ordinate plot illustrated in Figure 85. Here all three p -level values for each of the selected technology combinations are plotted simultaneously on a parallel co-ordinate plot. This plot shows the relative difference between two solutions in all objective dimensions. It can be noticed here that combination A has lower uncertainties than combination B in almost all objective dimensions. Based on this observation, the technology combination with lower uncertainty should be selected.

This solution was arrived at after making certain implicit tradeoffs and deciding on preferences. A different set of technology combination may have been selected

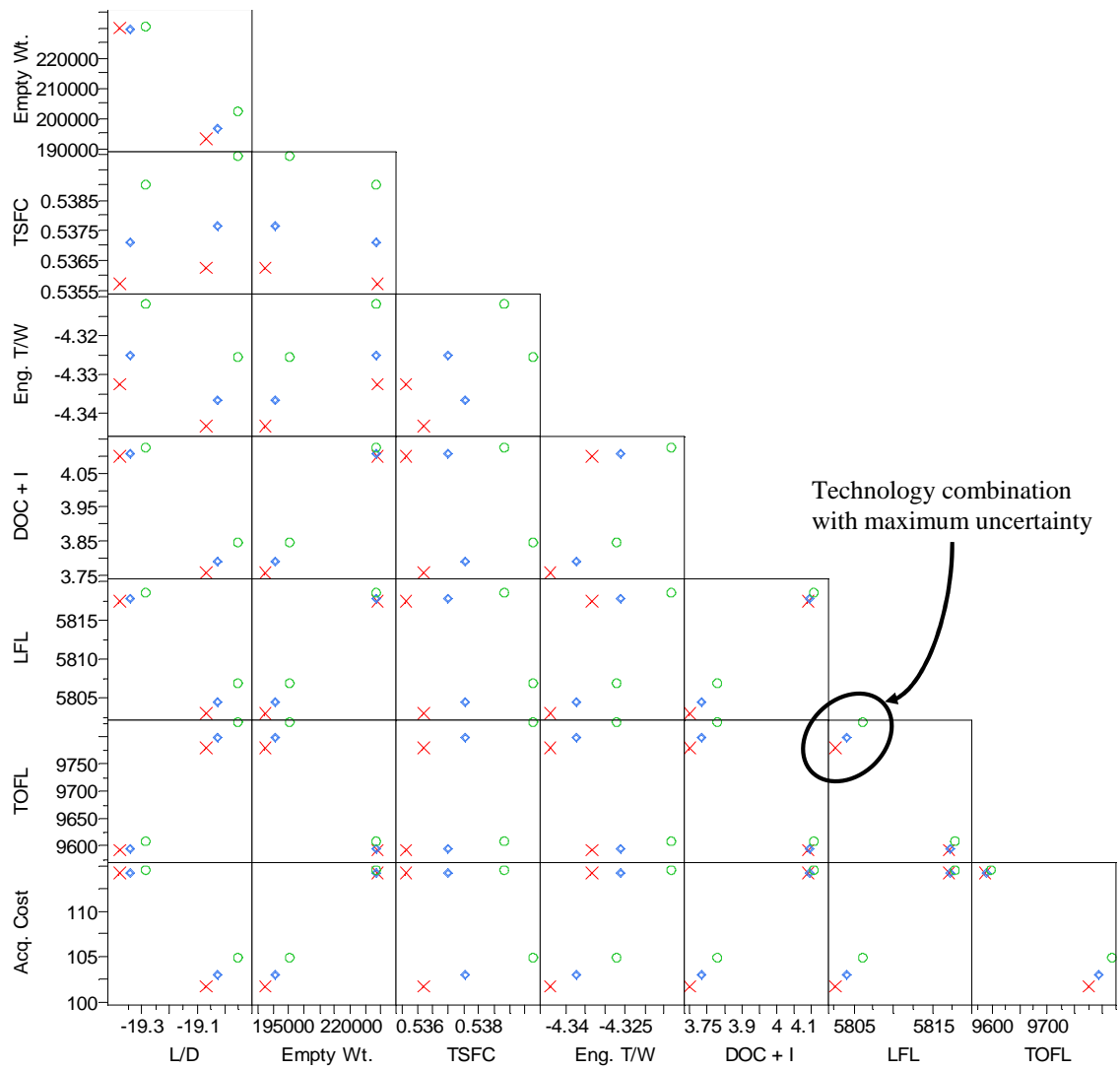


Figure 84: Scatter Plot Matrix with Selected Technology Combinations

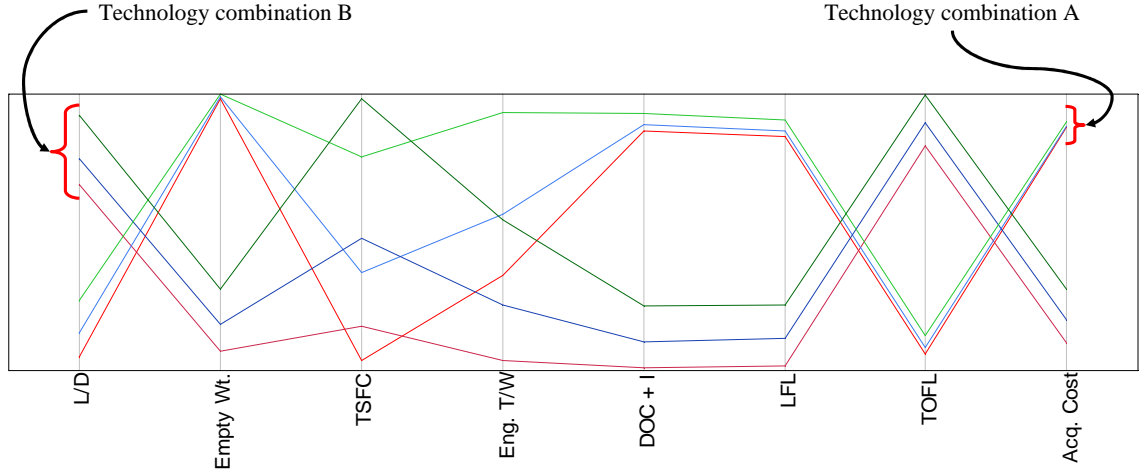


Figure 85: Parallel Co-ordinate Plot for Selected Technology Combinations

with these strategies but with different tradeoffs and preferences. But for any final solution, its relative position in the design space is immediately known.

9.9 Summary

The purpose of this chapter was to illustrate via an example the application of Pareto Optimization and Selection (POST) method developed in the previous chapter. Various steps involved in POST are applied on a technology exploration problem for a 300 passenger commercial aircraft. When the number of technology options is high, as in case of the example problem with 29 technologies, complete evaluation of the permissible combinations is not an option. To address this a stochastic algorithm for finding Pareto optimal solutions was implemented. The dimensionality of the objective space was also large with 11 objectives to minimize and 4 constraints to satisfy. It was decided to reduce this dimensionality using a dominance based dimensionality reduction. This method tries to maintain the relative structure of the Pareto frontier while identifying redundant objectives. After this procedure, eight most important objectives were selected. Moreover, there were uncertainties associated with the technology impacts. To address this, probabilistic Pareto optimization algorithm

was implemented. The result of this procedure was a set of 1500 Pareto optimal technology combinations in an eight dimensional objective space. Each combination had three values associated with a response. These three values correspond to three probability levels considered. Thus three Pareto layers were formed representing 50%, 75%, and 95% probability levels. These points were transferred to JMP[®] for visual exploration of combinatorial space. It was shown that visual exploration and decision making is not an easy task and some strategies were adopted for systematic exploration. The availability of different Pareto layers opened up the possibility of making tradeoffs in yet another dimension of uncertainty. It was demonstrated that apart from facilitating decision making with confidence, POST method also allows for systematic exploration of the entire combinatorial technology space; first using Pareto optimizing stochastic algorithm and in the final step via visual exploration.

CHAPTER X

CONCLUSIONS

Technology selection for a complex system is a challenging exercise in multi-objective combinatorial optimization. There are constraints to satisfy and many objectives to meet, in addition to the complexity of the problem that increases exponentially with increasing technologies. It has been observed in this document that these challenges can be successfully addressed by implementing proper tools and techniques. A systematic process can help identify efficient technology combinations out of many. Designers and decision makers can then explore these efficient or Pareto optimal combinations and select the most appropriate one. This chapter lists some of the key contributions made to the field of technology selection while developing this process. Some recommendations are made based on the lessons learned and ideas for extending this work are presented.

10.1 Summary of Contributions

Some of the key characteristics required for a technology selection process were identified at the beginning of this thesis. The concentration was on multi-objectivity of the problem and the presence of uncertainties. A technology selection *advisor* was also desired to advise on methods and algorithms to be used for a wide array of technology selection problems. Throughout the thesis a number of questions were posed and hypotheses proposed structuring the research.

10.1.1 Technology Selection Advisor

The first **question** asked: *What is the state of the art in technology selection process?* Chapter 2 and 3 were focused towards addressing this question. A review of

current literature showed that most of the methods focused on optimizing technology combinations based on a single objective. If multiple objectives were present, they were combined in the form of a utility function and this function was later optimized. Various approximate and exact algorithms for optimization were reviewed and demonstrated on a knapsack problem. Some exploratory techniques were also demonstrated. Based on this study of different algorithms and techniques for technology selection, framework for a technology selection advisor was presented. Some of the limitations in these techniques were identified and further research focused on addressing them.

10.1.2 Multi-Objective Technology Decisions

One of the most significant limitations of current methods was their handling of multiple objectives. Thus, the second research **question** was: *How to address multi-objective nature of the problem?* Discussion in Chapter 4 and 5 was focused towards answering this question. To investigate the multi-objective aspect and various options available for addressing it a detailed study of multi-objective decision making approaches was conducted. Specific reasons for the unsuitability of weighted sum or utility function approach were discussed. In light of these observations, a high level hypothesis was proposed:

Hypothesis A: A Posteriori preference articulation, a class of MODM methods, can be used to address multiple objectives in the technology selection problem and identify a satisfactory solutions.

A posteriori preference articulation includes within its scope the concept of Pareto optimality. Here, a set of Pareto optimal solutions is presented to the decision makers. They then decide on the best solution based on their preferences and by making implicit and explicit tradeoffs. Various aspects of Pareto optimality were discussed for the purpose and challenges identified. One of the main challenges involved was

with the dimensionality of the problem. The focus now was on reducing the problem dimensionality in the context of Pareto optimization and identifying an algorithm for Pareto optimization. It was hypothesized that:

Hypothesis B: Dimensionality of the Pareto hyper-surface in a multi-objective technology selection problem will be smaller than the number of objectives.

The plausibility of this hypothesis was checked on the technology problem with the help of dominance based dimensionality reduction technique. It was shown that the initial set of 11 objectives can be reduced to a set of eight objectives by accepting some error in the dominance structure. For this example, no objective was completely redundant in terms of dominance structure. Though, if all the 15 responses were considered as objectives to be minimized and dimensionality reduction applied to this space, two objectives were found to be redundant and their absence would not change the structure of the Pareto front.

For the purpose of Pareto optimization, various stochastic evolutionary algorithms were reviewed and a hypothesis was proposed:

Hypothesis C: Pareto optimization of the technology selection problem can be most efficiently accomplished by the Strength Pareto Evolutionary Algorithm.

The plausibility of this was checked using the benchmark knapsack problem. SPEA2 was demonstrated to be very effective in finding non-dominated solutions when compared to Niche Pareto genetic algorithm or a random search. The quality of results from SPEA2 was largely dependent on the archive size. Large archive populations helped achieve better results.

10.1.3 Probabilistic Pareto Optimization

The next research **question** asked: *How to account for technological uncertainties while selecting technology combinations?* Chapter 6 explored ideas towards answering this question. Different facets of uncertainty-based design were looked into and

probabilistic techniques explored for analysis. Uncertainty representation for technology impacts was implemented by converting expert opinion into a generalized beta distribution using PERT approximations. A framework based on response surface methodology and Monte Carlo simulations was selected for probabilistic analysis. The number of samples required by MCS to create a sufficiently accurate empirical CDF was calculated. Joint probability distribution was studied and the use of marginal distributions to account for uncertainties in each response was preferred. It was illustrated that post-optimization probabilistic analysis was not sufficient and the results would be different if the optimization was carried out based on probabilistic analysis. To address this concern within the framework of *a posteriori* preference articulation a hypothesis was proposed:

Hypothesis D: An approach based on probabilistic Pareto *layers* is the most appropriate method of accounting for technological uncertainties within the MODM framework of *a posteriori* preference articulation.

A novel probabilistic Pareto optimization algorithm based on SPEA2 was formulated to take into account the probabilistic values of the responses. This algorithm is designed to simultaneously search for non-dominated points on different Pareto layers and do this while taking into consideration the distribution widths of various points. To check this method and the plausibility of the previous hypothesis, the results of deterministic and probabilistic Pareto optimization of a knapsack problem were compared. It was demonstrated that probabilistic Pareto optimizer indeed provided better results than the deterministic optimizer.

10.1.4 Technology Incompatibilities

When there is a large number of technology options, there may be compatibility or enabling interactions among them. That is, one technology may not work in the presence of the other, or one will require the presence of the other technology. The

analysis of technology incompatibilities was implemented using the concepts of graph theory in Chapter 7. It was shown that this type of constraints significantly reduce the technology combinatorial space. An equation was derived for the average complexity of this space based on the number of technologies and constraints present. Based on this number, it can be decided if the complete evaluation of technology combinations is feasible or not. Given the fact that the average technology problem would have only about 10% permissible combinations, a hard constraint formulation based on gene correction is used in the Pareto optimizer.

10.1.5 Combinatorial Technology Space Exploration

Based on the ideas discussed above, a method called Pareto Optimization and Selection of Technologies (POST) was formulated. At the core, this method is about searching the combinatorial technology space to determine the probabilistic Pareto layers; and then, visually exploring those Pareto layers. The designers and decision makers would be able to know the limits of the objective space opened up by the technology options and also find the most suitable technology combination. Implicit tradeoffs can be made not only among various objectives but also with uncertainties. Once the Pareto layers and exploration environment are created for a problem, the decisions can be revisited if the preferences or constraints change over time without having to run time consuming analysis again.

The POST method and the codes developed are modular in nature. Various components can be replaced if something better comes along the way. For example, if a better probabilistic analysis technique is available, it can be easily used in place of MCS given its output is in the same format. If sufficient computational resources are available, complex system codes can easily replace the response surface equations used in the current example. Similarly, a custom built visualization environment can also be used in place of JMP[®]. The unique feature of this method that cannot be

replaced is the probabilistic Pareto optimization algorithm. This has been specifically developed to create probabilistic Pareto layers in the objective space in a single run.

10.2 Recommendations and Future Work

The concepts and techniques discussed above were applied to two example problems. In Chapter 1, the similarity of technology problem with the knapsack problem was explored. Based on this, a 5-dimensional 16 item knapsack problem was created as a benchmark problem to experiment various techniques studied. This proved to be invaluable in exploring various algorithms that can be used for technology selection. As the problem was small enough, a complete enumeration of all item combinations was possible and a *true* Pareto front was extracted. This proved to be very useful in comparing different Pareto optimization algorithms.

The knapsack problem can be considered a simplified version of the technology selection problem. The scale of a practical technology problem is much larger than the benchmark problem considered. Moreover, there are many intricacies involved in the technology problem because of their complex interactions with the system. Thus, an approach resulting from the experiments on the knapsack problem had to be implemented on a larger and practical technology exploration and selection problem. This problem was of technology exploration for a 300 passenger commercial aircraft.

The procedure for dimensionality reduction was applied on this problem. The main reason for applying this was for improving the efficiency of Pareto optimization algorithm and for the ease of visual exploration and decision making. Because of this, some objectives were not available to make tradeoffs. Even though these dimensions had minimum impact on the Pareto front, their absence can be construed as a drawback in certain cases. The availability of probabilistic Pareto layers adds significant capability to the combinatorial exploration process. Each Pareto layer can be explored individually based on the risk avoidance preferences of the decision

makers. The technology combinations can be down-selected based on the distance between their corresponding points in different Pareto layers. It was observed toward the end of the process that even though POST is a good tool for visual exploration, the decision making is still a daunting task. This is especially true for a very large dimensional problem without many constraints. Integrating a multi-attribute decision making (MADM) technique within this visual environment can prove useful for the purpose.

10.2.1 Ideas for Further Research

The concepts and framework developed in this research lead to some exciting new ideas for technology exploration and selection. For example, the technologies considered in this research had a point impact on the system. That is, there was only one value, though uncertain, that represented the impact of certain technology on a few technology metrics or k -factors. In many practical situations, this may not be true and the technology impact is more like a curve on a 2-dimensional k -factors plot. There is an initial tradeoff involved in deciding the impact of technology on two k -factors. For example, consider a case of thermal barrier coating technology under consideration for an aircraft engine turbine blades. By applying this technology, the turbine inlet temperature can be increased and at the same time cooling flow for the turbine blades can also be decreased. The impact on these two k -factors is interdependent. It would be of value to bring in this type of information when evaluating and selecting technology combinations. One way of implementing this within the Pareto optimization framework is via discretizing the technology curve. Thus each technology is represented by n points, each with different values for k -factors. For a smaller problem with only a few technologies, a gradient based optimizer can be used to fix the values of k -factors within a branch and bound algorithm for optimizing technology combinations.

This research addressed the probabilistic nature of technology impacts. But, the distributions on these impacts change with time as more information is available about the technology and the system. From this perspective, they are stochastic in nature. One idea that may address this is by mapping the changing distributions over time to the generations of probabilistic Pareto optimization algorithm. In this way, the first generation will use the initial distribution on the technology impact and the last generation would use the final predicted distribution. Implementing this idea would require extensive research in verifying the concept of mapping generations to the distributions.

10.3 In Closing

The main objective of this research was to formulate a method which helps designers and decision makers efficiently explore the technology combinatorial space and select the most appropriate combination. The multi-dimensional and uncertain nature of the problem was to be considered. Achieving this objective required formulating a novel probabilistic Pareto optimization algorithm that would result in probabilistic Pareto layers of technology combinations in the objective space. Using various techniques, a method is developed that brings in the results from a consistent analytical foundation in a dynamic tradeoff environment where the decision makers can make implicit tradeoffs and select technologies while being aware of all the options available and the risk involved.

REFERENCES

- [1] Daniel P. Raymer. *Aircraft Design: A Conceptual Approach*. AIAA, third edition, 1999.
- [2] Dimitri N. Mavris, Andrew P. Baker, and Daniel P Schrage. IPPD through robust design simulation for an affordable short haul civil tiltrotor. Paper 1997, AHS, 1997.
- [3] Best practices: Better matching of needs and resources will lead to better weapon system outcomes. Technical Report GAO-01-228, U. S. Government Accountability Office, March 2001.
- [4] Obaid Younossi, David E. Stem, Mark A. Lorell, and Frances M. Lussier. Lessons learned from the F/A-22 and F/A-18E/F development programs. Technical Report MG-276, RAND, 2005.
- [5] Tactical aircraft: DOD needs a joint and integrated investment strategy. Technical Report GAO-07-415, U.S. Government Accountability Office, April 2007.
- [6] Best practices: Using a knowledge-based approach to improve weapon acquisition. Technical Report GAO-04-386SP, U.S. Government Accountability Office, January 2004.
- [7] Best practices: Capturing design and manufacturing knowledge early improves acquisition outcomes. Technical Report GAO-02-701, U. S. Government Accountability Office, 2002.
- [8] John C. Mankins. Technology readiness levels. A White Paper, April 1995. Advanced Concept Office, Office of Space Access and Technology, NASA.
- [9] Best practices: Better management of technology can improve weapon system outcome. Technical Report GAO/NSIAD-99-162, U. S. Government Accountability Office, July 1999.
- [10] Defense acquisitions: Assessments of selected major weapon programs. Technical Report GAO-06-391, U. S. Government Accountability Office, March 2006.
- [11] Best practices: Stronger practices needed to improve dod technology transistion processes. Technical Report GAO-06-883, U. S. Government Accountability Office, September 2006.
- [12] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Co., 1979.

- [13] Michelle R. Kirby. *A Method for Technology Identification, Evaluation and Selection in Conceptual and Preliminary Aircraft Design*. PhD thesis, Georgia Institute of Technology, Atlanta, GA 30332, 2001.
- [14] Michelle R. Kirby and Dimitri N. Mavris. A technique for selecting emerging technologies for a fleet of commercial aircraft to maximize R&D investment. SAE Paper 2001-01-3018, 2001.
- [15] Dimitri N. Mavris and Michelle R. Kirby. Technology identification, evaluation and selection for commercial transport aircraft. Paper SAWE 2456, 58th Annual Conference Of Society of Allied Weight Engineers, San Jose, CA, May 1999.
- [16] Dimitri N. Mavris, Andrew P. Baker, and Daniel P. Schrage. Technology infusion and resource allocation for a civil tiltrotor. Paper AHS-2000-01-AB, American Helicopter Society, Inc., San Francisco, CA, January 2000.
- [17] Lou Cohen. *Quality Function Deployment : How to Make QFD Work for You*. Addison-Wesley, 1995.
- [18] Michael Brassard and Diane Ritter. *The Memory Jogger II*. GOAL/QPC, Salem, NH, first edition, 1994.
- [19] G. E. P. Box and K. B. Wilson. On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society*, 13(1):1–45, 1951.
- [20] Clay A. Ward. Technology tracking for the high speed research program. Technical Report SAE 985549, AIAA and SAE, Anaheim, CA, September 1998.
- [21] Michelle R. Kirby, Dimitri N. Mavris, and Matthew C. Largent. A process for tracking and assessing emerging technology development programs for resource allocation. In *1st AIAA Aircraft Technology Integration and Operations Forum*, number 2001-5280, Los Angeles, CA, October 2001. AIAA.
- [22] Michelle R. Kirby, Chris Raczynski, and Dimitri N. Mavris. An approach for strategic planning of future technology portfolio. AIAA Paper 2006-7788, AIAA, 2006.
- [23] Charles R. Weisbin, Guillermo Rodriguez, Alberto Elfes, and Jeffrey H. Smith. Toward a systematic approach for selection of nasa technology portfolios. *Systems Engineering Journal*, 7(4):285–302, 2004.
- [24] A. Elfes, C. R. Weisbin, R. Manvi, V. Adumitroaie, W. P. Lincoln, and K. Shelton. Extending the start framework: Computation of optimal capability development portfolios using a decision theory approach. *Systems Engineering*, 9(4):331–357, 2006.
- [25] Ram Manvi, Charles Weisbin, Wayne Zimmerman, and Guillermo Rodriguez. Decision tree assessment for challenging technologies for mission to europa. *Journal of Aerospace Engineering*, 16(3):121–128, July 2003.

- [26] Alberto Elfes, William P. Lincoln, Guillermo Rodriguez, Charles R. Weisbin, and Julie A. Wertz. Risk based technology portfolio optimization for early space mission design. In *Transactions on Aerospace and Electronic Systems*, volume 42. IEEE, January 2006.
- [27] Virgil Adumitroaie, Kacie Shelton, Alberto Elfes, and Charles R. Weisbin. Postoptimality analysis in the selection of technology portfolios. In *Conference on Systems Engineering Research*, Los Angeles, CA, April 2006. INCOSE.
- [28] Robert Shishko, Donald H. Ebbeler, and George Fox. NASA technology assessment using real options valuation. *Systems Engineering*, 7(1), 2004.
- [29] Juliana Hsuan Mikkola. Portfolio management of r&d projects: Implications for innovation management. *Technovation*, 21:423–435, 2001.
- [30] Rias J. Van Wyk. Strategic technology scanning. *Technological Forecasting and Social Change*, 55:21–38, 1997.
- [31] Keun-Tae Cho and Cheol-Shin Kwon. Hierarchies with dependence of technological alternatives: A cross-impact hierarchy process. *European Journal of Operational Research*, 156:420–432, 2004.
- [32] Alan L. Porter, W. Bradford Ashton, Guenter Clar, Joseph F. Coates, et al. Technology futures analysis: Toward integration of the field and new methods. *Technological Forecasting & Social Change*, 71:287–303, 2004. By Technology Futures Analysis Methods Working Group.
- [33] Aditya Utturwar, Sriram Rallabhandi, Daniel A. DeLaurentis, and Dimitri N. Mavris. A two-step optimization approach for technology selection. *Engineering Optimization*, 38(8):889 – 908, December 2006.
- [34] Karl Doerner, Walter J. Gutjahr, RICHARD F. HARTL, CHRISTINE STRAUSS, and CHRISTIAN STUMMER. Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. *Annals of Operation Research*, 131:79–99, 2004.
- [35] Frederic Villeneuve. *A Method for Concept and Technology Exploration of Aerospace Architectures*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, 2007.
- [36] Hongyi Sun and Tianchao Ma. A packing-multiple-boxes model for r&d project selection and scheduling. *Technovation*, 25:1355–1361, 2005.
- [37] Raj Subbu, Piero P. Bonissone, Neil Eklund, Srinivas Bollapragada, and Kete Chalermkraivuth. Multiobjective financial portfolio design: A hybrid evolutionary approach. In *IEEE Congress on Evolutionary Computation*, number 1722-1729, 2005.

- [38] Silvano Martello and Paolo Toth. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley and Sons, 1990.
- [39] David Pisinger. *Algorithms for Knapsack Problems*. PhD thesis, Dept. of Computer Science, University of Copenhagen, Universitetsparken 1, DK-2100 Copenhagen, Denmark, February 1995.
- [40] Sanjoy Dasgupta, Christos H. Papadimitriou, and Umesh Vazirani. *Algorithms*. McGraw-Hill Higher Education, 2006.
- [41] Toshihide Ibaraki. Design and analysis of approximate algorithms. *Annals of Operations Research*, 11:441–510, 1987.
- [42] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. MIT Press, Cambridge, MA, U.S.A., 1999.
- [43] George B. Dantzig. Discrete variable extremum problems. *Operations Research*, 5:266–277, 1957.
- [44] Egon Balas and Eitan Zemel. An algorithm for large zero-one knapsack problems. *Operations Research*, 28(5):1130–1154, 1980.
- [45] Peter J. Kolesar. A branch and bound algorithm for the knapsack problem. *Management Science*, 13(9):723–735, 1967.
- [46] Ellis Horowitz and Sartaj Sahni. Computing partitions with applications to the knapsack problem. *Journal of ACM*, 21(2):277–292, 1974.
- [47] Silvano Martello and Paolo Toth. An upper bound for the zero-one knapsack problem and a branch and bound algorithm. *European Journal of Operational Research*, 1(3):169–175, May 1977.
- [48] Harold Greenberg and Robert L. Hegerich. A branch search algorithm for the knapsack problem. *Management Science*, 16(5):327–332, January 1970.
- [49] The MathWorks, Inc. *MATLAB Optimization Toolbox 4 User's Guide*, March 2008. Online Only.
- [50] Richard Bellman. Dynamic programming and the numerical solution of variational problems. *Operations Research*, 5:277–288, 1957.
- [51] Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack Problems*. Springer, 2004.
- [52] Douglas C. Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons, 6 edition, 2004.
- [53] SAS Institute Inc., Cary, NC, USA. *JMP User Guide*, 7 edition, 2007.

- [54] W. J. Krzanowski. *Principles of Multivariate Analysis: A User's Perspective*. Oxford Statistical Science Series. Oxford University Press Inc., Revised edition, 2000.
- [55] Bryan F.J. Manly. *Multivariate Statistical Methods : A Primer*. Chapman and Hall/CRC, 3rd edition, 2004.
- [56] David A. Van Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analysis, and New Innovations*. PhD thesis, Airforce Institute of Technology, 1996.
- [57] Ching-Lai Hwang and Abu Syed Md. Masud. *Multiple Objective Decision Making - Methods and Applications*. Number 164 in Lecture Notes in Economics and Mathematical Systems. Springer-Verlag, 1979.
- [58] I. Das and J. E. Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural Optimization*, 14:63–69, 1997.
- [59] Jeffrey Horn. *The Nature of Niching: Genetic Algorithms and the Evolution of Optimal, Cooperative Populations*. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL 61801, 1997.
- [60] George A. Hazelrigg. Validation of engineering design alternative selection methods. *Engineering Optimization*, 35(3):103–120, April 2003.
- [61] Ian C. Parmee, Dragan Cvetković, Andrew H. Watson, and Christopher R. Bonham. Multiobjective satisfaction within an interactive evolutionary design environment. *Evolutionary Computation*, 8(8):197–222, 2000.
- [62] Michael Alexander Buonanno. *A Method for Aircraft Concept Exploration Using Multicriteria Interactive Genetic Algorithms*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, USA, 2005.
- [63] Carlos A. Coello Coello. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems*, 1(3):129–156, 1999.
- [64] Kalyanmoy Deb and Dhish Kumar Saxena. On finding pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. Technical Report 2005011, Kanpur Genetic Algorithms Laboratory, Kanpur, India, December 2005.
- [65] Dimo Brockhoff and Eckart Zitzler. Are all objectives necessary? on dimensionality reduction in evolutionary multiobjective optimization. In *Parallel Problem Solving from Nature - PPSN IX*, volume 4193 of *Lecture Notes in Computer Science*, pages 533–542. Springer, 2006.

- [66] Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, November 1999.
- [67] Eckart Zitzler. Evolutionary algorithms for multiobjective optimization. In K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papailiou, and T. Fogarty, editors, *Evolutionary Methods for Design, Optimisation and Control*, Barcelona, Spain, 2002. CIMNE.
- [68] Ludovic Lebart, Alain Morineau, and Kenneth M. Warwick. *Multivariate Descriptive Statistical Analysis: Correspondence Analysis and Related Techniques for Large Matrices*. Wiley Series in Probability and Statistics. John Wiley & Sons, 1984. Translated by: Elisabeth Moraillon Berry.
- [69] Dimo Brockhoff and Eckart Zitzler. On objective conflicts and objective reduction in multiple criteria optimization. Report 243, TIK, ETH Zurich, Zurich, Switzerland, February 2006.
- [70] Dimo Brockhoff and Eckart Zitzler. Dimensionality reduction in multiobjective optimization with (partial) dominance structure preservation: Generalized minimum objective subset problems. Report 247, TIK, ETH Zurich, Zurich, Switzerland, April 2006.
- [71] Bryce Roth, Dimitri N. Mavris, Mathew D. Graham, and Noel Macsotai. Adaptive selection of pareto-optimal engine technology solution sets. Paper 2002–594, ICAS, 2002.
- [72] I. Das and J. E. Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998.
- [73] L. Cheng, E. Subrahmanian, and A.W. Westerberg. Design and planning under uncertainty: issues on problem formulation and solution. *Computers and Chemical Engineering*, 27:781–801, 2003.
- [74] J. Rakowska, R. T. Haftka, and L. T. Watson. Tracing the efficient curve for multi-objective control-structure optimization. *Computing Systems Engineering*, 2(5/6):461–471, 1991.
- [75] A. Chakraborty, D. C. S. Allison, C. J. Ribbens, and L. T. Watson. Parallel unit tangent vector computation for homotopy curve tracking on a hypercube. In *Proceedings of the 1990 ACM annual conference on Cooperation*, pages 103–108, Washington D.C., USA, 1990. ACM.
- [76] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

- [77] Carlos A. Coello Coello. Evolutionary multi-objective optimization: A critical review. In Ruhul Sarker, Masoud Mohammadian, and Xin Yao, editors, *Evolutionary Optimization*, International Series in Operations Research and Management Science, chapter 5. Kluwer Academic Publishers, 2002.
- [78] David A. Van Veldhuizen and Gary B. Lamont. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evolutionary Computation*, 8(2):125–147, 2000.
- [79] William M. Spears, Kenneth A. DeJong, Thomas Bäck, David B. Fogel, and Hugo deGaris. An overview of evolutionary computation. In Pavel B. Brazdil, editor, *Proceedings of the European Conference on Machine Learning (ECML-93)*, volume 667, pages 442–459, Vienna, Austria, 1993. Springer Verlag.
- [80] Xin Yao. Evolutionary computation: A gentle introduction. In Ruhul Sarker, Masoud Mohammadian, and Xin Yao, editors, *Evolutionary Optimization*, International Series in Operations Research and Management Science, chapter 2. Kluwer Academic Publishers, 2002.
- [81] Darrell Whitley. An overview of evolutionary algorithms: Practical issues and common pitfalls. *Information and Software Technology*, 43(14):817–831, 2001.
- [82] Sami Khuri, Thomas Bäck, and Jörg Heitkötter. An evolutionary approach to combinatorial optimization problems. In D. Cizmar, editor, *Proceedings of the 22nd Annual ACM Computer Science Conference*, pages 66–73, New York, 1994. ACM Press.
- [83] Sami Khuri, Thomas Bäck, and Jörg Heitkötter. The zero/one multiple knapsack problem and genetic algorithms. In E. Deaton, D. Oppenheim, J. Urban, and H. Berghel, editors, *ACM Symposium of Applied Computation*, pages 188–193. ACM Press, 1994.
- [84] A.E. Eiben, J.K. van der Hauw, and J.I. van Hemert. Graph coloring with adaptive evolutionary algorithms. *Journal of Heuristics*, 4(1):25–46, June 1998.
- [85] Jens Gottlieb, Elena Marchiori, and Claudio Rossi. Evolutionary algorithms for the satisfiability problem. *Evolutionary Computation*, 10(1):35–50, March 2002. MIT Press.
- [86] Eric Michielssen and Daniel S. Weile. Electromagnetic system design using genetic algorithms. In G. Winter, J. Périaux, M. Galán, and P. Cuesta, editors, *Genetic Algorithms in Engineering and Computer Science*, chapter 18, pages 344–369. John Wiley and Sons, 1995.
- [87] Alice E. Smith and Bryan A. Norman. Design of production facilities. In Ruhul Sarker, Masoud Mohammadian, and Xin Yao, editors, *Evolutionary Optimization*, International Series in Operations Research and Management Science, chapter 12. Kluwer Academic Publishers, 2002.

- [88] Eloy Díaz-Dorado, José Cidrás, and Edelmiro Míguez. Application of evolutionary algorithms for the planning of urban distribution networks of medium voltage. In *IEEE Transactions on Power Systems*, volume 17, pages 879–884, August 2002.
- [89] Nicholas J. Radcliffe. Non-linear genetic representations. In Reinhard Männer and Bernard Manderick, editors, *Parallel problem solving from nature 2*, pages 259–268, Amsterdam, 1992. Elsevier.
- [90] Joshua D. Knowles. *Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization*. PhD thesis, University of Reading, Reading, RG6 6AY, UK, January 2002.
- [91] P.C. Chu and J. E. Beasley. A genetic algorithm for multidimensional knapsack problem. *Journal of Heuristics*, 4:63–86, 1998. Kluwer Academic Publishers.
- [92] Zbigniew Michalewicz. A survey of constraint handling techniques in evolutionary computation methods. In John R. McDonnell, Robert G. Reynolds, and David B. Fogel, editors, *4th Annual Conference on Evolutionary Programming*, pages 135–155, Cambridge, MA, 1995. MIT Press.
- [93] Carlos A. Coello Coello. A survey of constraint handling techniques used with evolutionary algorithms. Technical Report Lania-RI-9904, Laboratorio Nacional de Informtica Avanzada, Veracruz 91090, Mexico, 1999.
- [94] Carlos A. Coello Coello. Theoretical and numerical constraint handling techniques used with evolutionary algorithms: A survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191(11–12):1245–1287, 2002.
- [95] Nicholas J. Radcliffe and Patrick D. Surry. Fundamental limitations on search algorithms: Evolutionary computing in perspective. In *Computer Science Today: Recent Trends and Developments*, pages 275–291. 1995.
- [96] David H. Wolpert and William G. Macready. No free lunch theorems for search. Technical Report SFI-TR-95-02-010, Santa Fe, NM, 1995.
- [97] David H. Wolpert and William G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, April 1997.
- [98] Thomas M. English. Evaluation of evolutionary and genetic optimizers: No free lunch. In L. J. Fogel, P. J. Angeline, and T. Bäck, editors, *Evolutionary Programming V: Proceedings of the Fifth Annual Conference on Evolutionary Programming*, pages 163–169, 1996.
- [99] Joseph C. Culberson. On the futility of blind search. Technical Report 96-18, Edmonton, Alberta, Canada, 1996.

- [100] J. David Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms*, pages 93–100, Mahwah, NJ, USA, 1985. Lawrence Erlbaum Associates, Inc.
- [101] Frank Kursawe. A variant of evolution strategies for vector optimization. In H. P. Schwefel and R. Männer, editors, *Parallel Problem Solving from Nature. 1st Workshop*, volume 496, pages 193–197, Berlin, Germany, 1991. Springer-Verlag.
- [102] Carlos A. Coello Coello. A short tutorial on evolutionary multiobjective optimization. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 21–40. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.
- [103] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.
- [104] Carlos M. Fonseca and Peter J. Fleming. Genetic algorithms for multiobjective optimizations: Formulation, discussion and generalization. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, San Mateo, CA, July 1993. Morgan Kaufmann.
- [105] N. Srinivas and Kalyanmoy Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.
- [106] Kalyanmoy Deb, Samir Agrawal, Amrit Pratab, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, J. J. Merelo, and Hans-Paul Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.
- [107] Jeffrey Horn, Nicholas Nafpliotis, and David E. Goldberg. Multiobjective optimization using the niched pareto genetic algorithm. IlliGAL Report 93005, University of Illinois at Urbana-Champaign, Urbana, IL 61801, July 1993. also, Published, in part, in The Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, Volume 1, 1994, pp. 82–87.
- [108] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the strength pareto evolutionary algorithm. Report 103, TIK, ETH Zurich, Zurich, Switzerland, September 2001.

- [109] Eckart Zitzler, Marcos Laumanns, and Lothar Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papailiou, and T. Fogarty, editors, *Evolutionary Methods for Design, Optimisation and Control*, Barcelona, Spain, 2002. CIMNE.
- [110] Thomas A. Zang, Michael J. Hemsch, Mark W. Hilburger, Sean P. Kenny, James M. Luckring, Peiman Maghami, Sharon L. Padula, and W. Jefferson Stroud. Needs and opportunities for uncertainty-based multidisciplinary design methods for aerospace vehicles. White Paper TM-2002-211462, NASA, Langley Research Center, Hampton, Virginia 23681-2199, July 2002.
- [111] Luc Huyse. Solving problems of optimization under uncertainty as statistical decision problems. Conference Paper A01-25252, AIAA, April 2001. Structural Dynamics and Materials Conference.
- [112] Wynn C. Stirling. *Satisficing Games and Decision Making: With Applications to Engineering and Computer Science*. Cambridge University Press, 2003.
- [113] J. C. Helton and W. L. Oberkampf. Alternative representations of epistemic uncertainty. *Reliability Engineering & System Safety*, 85(1-3):1–10, 2004.
- [114] William L. Oberkampf, Jon C. Helton, Cliff A. Joslyn, Steven F. Wojtkiewicz, and Scott Ferson. Challenge problems: uncertainty in system response given uncertain parameters. *Reliability Engineering & System Safety*, 85(1-3):11–19, 2004.
- [115] George J. Klir. Generalized information theory: aims, results, and open problems. *Reliability Engineering & System Safety*, 85(1-3):21–38, 2004.
- [116] J. C. Helton, J. D. Johnson, and W. L. Oberkampf. An exploration of alternative approaches to the representation of uncertainty in model predictions. *Reliability Engineering & System Safety*, 85(1-3):39–71, 2004.
- [117] Han Gil Chae. *A Possibilistic Approach to Rotorcraft Design Through a Multi-Objective Evolutionary Algorithm*. PhD thesis, Georgia Institute of Technology, December 2006.
- [118] Anthony O’Hagan and Jeremy E. Oakley. Probability is perfect, but we can’t elicit it perfectly. *Reliability Engineering & System Safety*, 85(1-3):239–248, 2004.
- [119] Robert G. Batson and Robert M. Love. Risk analysis approach to transport aircraft technology assessment. *Journal of Aircraft*, 25(2):99–105, February 1988.
- [120] Anthony J. Hayter. *Probability and Statistics*. PWS Publishing Company, 1996.

- [121] Arjun K. Gupta and Saralees Nadarajah, editors. *Handbook of Beta Distribution and its Applications*. Statistics. Marcel Dekker, Inc., 2004.
- [122] Harold A. Linstone and Murray Turoff, editors. *The Delphi Method : Techniques and Applications*. Addison-Wesley, 1975.
- [123] D. G. Malcolm, J. H. Roseboom, C. E. Clark, and W. Fazar. Application of a technique for research and development program evaluation. *Operations Research*, 7(5):646–669, 1959.
- [124] C. Perry and I. D. Greig. Estimating the mean and variance of subjective distributions in pert and decision analysis. *Management Science*, 21(12):1477–1480, 1975.
- [125] Donald L. Keefer and Samuel E. Bodily. Three-point approximations for continuous random variables. *Management Science*, 29(5):595–609, 1983.
- [126] David G. Robinson. A survey of probabilistic methods used in reliability, risk and uncertainty analysis: Analytical techniques i. Report SAND98-1189, Sandia National Laboratories, June 1998.
- [127] Y. T. Wu. *NESSUS/FPI Theoretical Manual*. Southwest Research Institute, November 1995. Prepared for NASA Lewis Research Center.
- [128] J. Sakamoto, Y. Mori, and T. Sekioka. Probability analysis method using fast fourier transform and its application. *Structural Safety*, 19(1):21–36, 1997.
- [129] Ravi C. Penmetsa and Ramana V. Grandhi. Adaptation of fast fourier transformations to estimate structural failure probability. *Finite Elements in Analysis and Design*, 39:473–485, 2003.
- [130] Y. T. Wu, H. R. Millwater, and T. A. Cruse. Advanced probabilistic structural analysis method for implicit performance functions. *AIAA Journal*, 28(9):1663–1669, 1990.
- [131] Eric P. Fox and Stefan Reh. On the accuracy of various probabilistic methods. Conference Paper A00-24743, AIAA, April 2000. Structural Dynamics and Materials Conference.
- [132] Paul F. Dienemann. Estimating cost uncertainty using monte carlo techniques. Memorandum RM-4854-PR, RAND Corporation, January 1966.
- [133] Oliver Bandte. *A probabilistic multi-criteria decision making technique for conceptual and preliminary aerospace systems design*. PhD thesis, Georgia Institute of Technology, Atlanta, Ga 30332, 2000.
- [134] Eric P. Fox. The pratt & whitney probabilistic design system. Conference Paper AIAA-94-1442, AIAA/ASME/ASCE/AHS/ASC, April 1994. Structures, Structural Dynamics, and Materials Conference.

- [135] Dimitri N. Mavris and Daniel A. DeLaurentis. A probabilistic approach for examining aircraft concept feasibility and viability. *Aircraft Design*, 3(2):79–101, 2000.
- [136] M.S. Eldred, H. Agarwal, V.M. Perez, S.F. Wojtkiewicz, and J.E. Renaud. Investigation of reliability method formulations in DAKOTA/UQ. *Structure and Infrastructure Engineering*, 3(3):199–213, 2007.
- [137] Dimitri N. Mavris, Oliver Bandte, and Daniel A. DeLaurentis. Robust design simulation: A probabilistic approach to multidisciplinary design. *Journal of Aircraft*, 36(1):298–307, 1999.
- [138] Apurva Kumar, Prasanth B. Nair, Andy J. Keane, and Shahrokh Shahpar. Robust design using bayesian monte carlo. *International Journal for Numerical Methods in Engineering*, 73(11):1497–1517, 2007.
- [139] Edward L. Melnick and Aron Tenenbein. Misspecifications of the normal distributions. *The American Statistician*, 36(4):372–373, November 1982.
- [140] Paul R. Garvey. A family of joint probability models for cost and schedule uncertainties. Conference Paper AD-A275513, 27th Annual Department of Defense Cost Analysis Symposium, September 1993.
- [141] Bryce Roth and Chirag Patel. Application of genetic algorithms in the engine technology selection process. *Journal of Engineering for Gas Turbines and Power*, 126(4):693–700, 2004. also ASME Paper 2003–GT–38482, June 2003.
- [142] Bryce Roth, Bryan J. German, Dimitri N. Mavris, and Noel Macsotai. Adaptive selection of engine technology solution sets from a large combinatorial space. Paper 2001–3208, AIAA, 2001.
- [143] Douglas B. West. *Introduction to Graph Theory*. Perintice-Hall Inc., second edition, 2005.
- [144] J. A. Bondy and U. S. R. Murty. *Graph Theory with Applications*. American Elsevier Publishing Co., 1976.
- [145] Dragan Stevanović, Vladimir Brankov, Dragoš Cvetković, and Slobodan Simić. newGRAPH: Integrated enviornment for graph theory. On-line resource: <http://www.mi.sanu.ac.yu/newgraph/index.html> Last Accessed: 11/15/2008.
- [146] Herbert S. Wilf. *Algorithms and Complexity*. Internet Edition, 1994. <http://www.math.upenn.edu/~wilf/AlgComp3.html> Last Accessed: 10/12/2008.
- [147] C. Bron and J. Kerbosch. Algorithm 457: Finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, 1973.

- [148] Christopher M. Raczynski, Jung-Ho Lewe, Michelle R. Kirby, and Dimitri N. Mavris. Technology portfolio assessments using a gene-correction genetic algorithm. AIAA Paper 2003-6731, AIAA, 2003.
- [149] Christopher M. Raczynski, Michelle R. Kirby, and Dimitri N. Mavris. Technology portfolio assessments using a multi-objective genetic algorithm. SAE Paper 2004-01-3144, SAE, Reno, NV, November 2004.
- [150] Dimitri N. Mavris, Michelle R. Kirby, and J. C. Tai. FY04 VISTA technology assessment. Project 1606X27, NASA Glenn Research Center, May 2005. Contract No. NAS 3-00179/NNL04AD06T.
- [151] Raymond H. Myers and Douglas C. Montgomery. *Response Surface Methodology: Process and Product Optimization Using Design Experiments*. John Wiley and Sons, Inc., New York, New York, 1995.
- [152] George E P Box and Norman R Draper. *Empirical Model-Building and Response Surfaces*. John Wiley & Sons, Inc., New York, NY, USA, 1986.
- [153] Dimo Brockhoff. Java implementations of exact and greedy objective reduction algorithms. <http://www.tik.ee.ethz.ch/sop/download/supplementary/objectiveReduction/> Last Accessed: 01/08/2009.
- [154] Dimitri N. Mavris, Patrick T. Biltgen, and Neil R. Weston. Advance design of complex systems using the collaborative visualization environment (COVE). Conference Paper AIAA 2005-126, AIAA, January 2005. Presented at 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada.
- [155] Jan Osburg and Dimitri Mavris. A collaborative design environment to support multidisciplinary conceptual systems design. In *Proceedings of the SAE World Aerospace Congress 2005*, number 2005-01-3435. SAE, 2005.

VITA

Chirag Patel was born in Delhi and grew up in Baroda, India. He went to North Maharashtra University, Shahada in August of 1996 and received his Bachelor's degree in Mechanical Engineering in June of 2000. He then enrolled in Aerospace System Design Laboratory (ASDL) at Georgia Institute of Technology and was awarded his Master's Degree in December of 2002. While at ASDL, he worked with a variety of aircraft engine manufacturers and government agencies. He passed his qualifying exam in April of 2003 and will be awarded his Ph.D in Aerospace Engineering in August of 2009. He currently resides in Orlando, FL with his family.