Infusion of Robustness into the Product Platform Constructal Theory Method

A Thesis
Presented to
The Academic Faculty

by

Rakesh S. Kulkarni

In Partial Fulfillment
of the Requirements for the Degree of
Master of Science in Mechanical Engineering

Georgia Institute of Technology
April 2005

Infusion of Robustness into the Product Platform Constructal Theory Method

Approved by:

Dr. Farrokh Mistree, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. David Rosen
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Janet Allen
School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: April 17, 2005

For my parents,

to whom I owe everything.

ACKNOWLEDGEMENTS

answering my questions regarding robustness. Thanks Greg, I will always remember all your jokes and happy moments spent together. Thank you Benay for helping me prepare my defense presentation. Thank you John for helping me out the previous night of my thesis submission. I want to thank Steve, Scott, John, Jay, Jamal, Bert, Chris, Nate, Andrew and Morgan for all the encouragement after the first catch of my yet to begin football career. Thank you, Stephanie and Jitesh for helping in the arrangements before my defense presentation.

I offer my deepest gratitude to my parents to whom I owe everything. I dedicate this thesis to them. I love you very much. I give heartfelt thanks to my brother, Rahul and his wife Madhura, who have lifted my spirits whenever I was down and encouraged me to do better. I would not be writing these lines without the unconditional love and support of my family.

I gratefully acknowledge all the help and guidance by my mentor at Xerox, Dr. Eduardo Bascaran. I would like to thank my roommates, Vishwanath, Sapan, Vivek, Manish and Joshy with whom I have spent some very enjoyable moments of my life. I acknowledge the financial support from NSF grant DMI-0085136 and DMI-0407627, and the Systems Realization Laboratory for underwriting the cost of computing time.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Today, mass customization has emerged as a manufacturing paradigm for a number of enterprises to efficiently and effectively satisfy customers' requirements for product variety. The competitive nature of today's market makes it necessary for designers to have a methodology for designing customized products in such a dynamic environment.

The Product Platform Constructal Theory Method (PPCTM), developed by Dr. Gabriel Hernandez, provides designers a methodical approach for synthesizing multiple modes of managing variety in the development of product platforms for customized products. The use of the PPCTM results in a hierarchical organization of the modes of managing customization, as well as the specification of their range of application across the product platform. The focus in this thesis is to augment the PPCTM in order to develop an effective product platform design method that alleviates three of its major limitations: inability to deal with uncertain distributions of demand, changing design parameters and changing extents of marketplaces. The infusion of concepts of robustness helps to address the first two limitations making the product platforms unaffected by large variations in demand and design parameters. The compromise Decision Support Problem is proposed to address the third limitation of changing extents of marketplaces by making tradeoffs between objectives of the initial market extent and future probable extensions.

The result of this work is an augmented PPCTM that facilitates the synthesis of multiple modes for managing product variety in the presence of a dynamic environment. The augmented method is used to design a line of customizable pressure vessels and hand exercisers.

CHAPTER I

THE CHALLENGES OF DESIGNING FOR MASS CUSTOMIZATION

In this thesis the author's objective is to augment the Product Platform Constructal Theory Method (PPCTM) in order to develop an effective product platform design method to provide a designer with the ability to synthesize multiple modes for managing customization in the development of product platforms for uncertain demand, changing design parameters and changing extents of marketplaces.

Specifically, the three augmentations for handling uncertain demand, changing design parameters and changing extents of marketplaces, are divided into two parts. In the first part, concepts of robustness are infused into the method. These concepts are used to give the designer the power to design product platforms that are robust to changes in demand and design parameters. In the second part, the use of compromise DSP to make tradeoffs between objectives of the initial as well as the future, potential, extended marketplaces is presented. This makes it possible to design platforms strategically that are effective in the initial as well as the future marketplaces.

The Product Platform Constructal Theory Method, developed by Gabriel Hernandez, provides designers a methodical approach for synthesizing multiple modes of managing customization in the development of product platforms for customized products (Hernandez et al., 2001). As a result of the PPCTM's theoretical foundations in both hierarchical systems theory and constructal theory, the design of platforms for

customizable products is represented as a problem of access in a geometric space. The result of the use of the PPCTM is a hierarchical organization of the modes of managing customization, as well as the specification of their range of application across the product platform.

1.1 BACKGROUND AND MOTIVATION – MASS CUSTOMIZATION

1.1.1　What is Mass Customization?

Joseph Pine II, author of *Mass Customization: The New Frontier In Business Competition*, defines mass customization as the production and distribution of customized goods and services on a mass basis. Pine quotes a Nissan Corporation manifesto of the Five A's: "Any volume, Any time, Anybody, Anywhere, and Anything." Simply stated, mass customization is about choice. It is about giving customers a unique end product when, where and how they want it. During the last couple of years, choice has become an important ingredient of consumer purchasing decisions.

Mass customization enables manufacturers to customize products quickly at a cost, efficiency and speed close to those of mass production. The core of mass customization is the ability to increase product variety and customization without corresponding increases in costs. This increase in product variety and customization is made possible through flexibility and quick responsiveness.

Mass customization makes it possible to manufacture products as per customer's desire or specifications, as opposed to producing a generic product to be placed in an inventory in hopes that some customer will later purchase it. The mass production cost curve and the mass customized production cost curve is in Figure 1.1. It is observed that

the difference of these curves from the curve that depicts the price that customers are willing to pay is significant. Mass production has an economic advantage over high volume production rather than in low volume production. However, the profits obtained by these products are relatively less than the profits obtained by using mass customized production. Thus, the capability to satisfy individual customer needs can translate into higher profit margins. This is the key advantage of pursuing mass customization, which enables manufacturing enterprises to satisfy their customers' demand for variety and at the same time minimize costs.
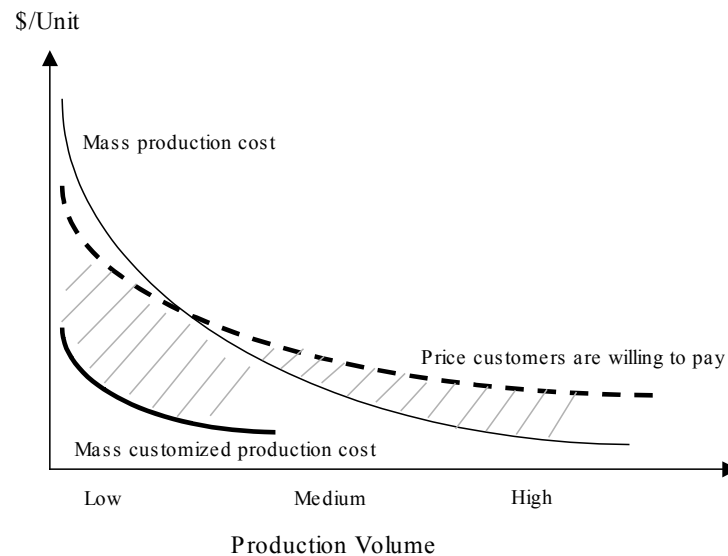
Figure 1.1: The Economic Implications of Mass Customization (Tseng et al., 1998)

1.1.2   Need of Mass Customization

Mass customization was born by the *convergence of need and capability*. Markets are becoming highly volatile because of changing customer needs, technological advances, and diminishing product life cycles. Today, most companies are looking for different business strategies to redefine themselves in this changing environment. In order

to have a competitive edge and survive in a global economy where customer diversity is extreme, companies have to provide "personalized" products for individual customers. Successful enterprises are those who can use these new capabilities to satisfy these new needs. All these factors are expected to be even more influential in the future. Tomorrow, enterprises in all branches of industry would be forced to react to the growing individualization of demand, yet, at the same time, increasing competitive pressure would dictate that costs must also continue to decrease. These reasons have made it more necessary for companies to embrace mass customization. It makes it possible to customize products quickly for individual customers or for niche markets at better than mass production efficiency and speed.

Due to better practices being used for mass customization, it has indirectly helped to reduce the cost of inventory, obsolescence, discounting, distribution, setup, equipment utilization, floor space, material overhead, and information systems. At the same time, the ability to give the customers what they want, when they want it enables the manufacturer to charge premium prices and thus earn more profits.

1.1.3 Challenges of Mass Customization

The challenges involved with designing a product and its manufacturing process for customization are inherent in the differences between mass production and mass customization. These differences are detailed in Table 1.1.

Table 1.1: The Differences between Mass Production and Mass Customization (Pine, 1993)

	Mass Production	*Mass Customization*
Focus	Efficiency through stability and control	Variety and customization through flexibility and quick responsiveness
Goal	Developing, producing, marketing, and delivering goods and services at prices low enough that nearly everyone can afford them	Developing, producing, marketing, and delivering affordable goods and services with enough variety and customization that nearly everyone finds exactly what they want
Key Features	Stable demand	Fragmented demand
	Large, homogeneous markets	Heterogeneous niches
	Low-cost, consistent quality, standardized goods, and services	Low-cost, high-quality, customized goods and services
	Long product development cycles	Short product development cycles
	Long product life cycles	Short product life cycles

The key differences outlined in Table 1.1 – mass customization's uncertain demand, heterogeneous niches, and short product life cycles – are some of the key challenges in the design of a customized part.

Firstly, uncertain demand poses a serious problem for design of products for customization. It is difficult to provide variety while being unsure of the demand of the large variety of products to be manufactured. Today enterprises need a production system that can adapt quickly to changing market conditions, provide the lowest costs, and give customers what they want, when they want it. The challenge for customization is to have techniques that are unaffected to varying demand of the products.

Secondly, one needs to interact with customers and understand what they want. This interaction not only helps in providing products that will satisfy the customers but also getting to know the trends of customers. The challenge is to provide products that satisfy the changing requirements of the customers. The change in the customer requirements may be in terms of change in the required design parameters. During the life cycle of a product line, there are many times when the design parameters need to be

changed to suit the customer requirements. It is not possible to have a product line that would be perfectly satisfying all the customer requirements during the entire life cycle of the product family. In such cases, it is always beneficial to have some of the design parameters of the product family that can be tweaked to satisfy the customer needs. One of the main objectives of being able to change design parameters is to increase the demand in the existing products by customizing the product according to exact customer specifications. Moreover, new markets can also be explored by such variations.

Thirdly, since product life cycles are shortening, it is important to have the transition of old products into new products easily. Moreover, capturing of new markets to satisfy the needs of the customer must be possible. In this thesis, marketplace is the space in which the manufacturer has demand for his products and wishes to sell his products in this space. This is the space in which the manufacturer designs his product family.

This need is better understood by analyzing the Figure 1.2. The figure helps to show the life cycle of a product family by plotting the market capture of variants of a product family over time.

Product Family Life Cycle
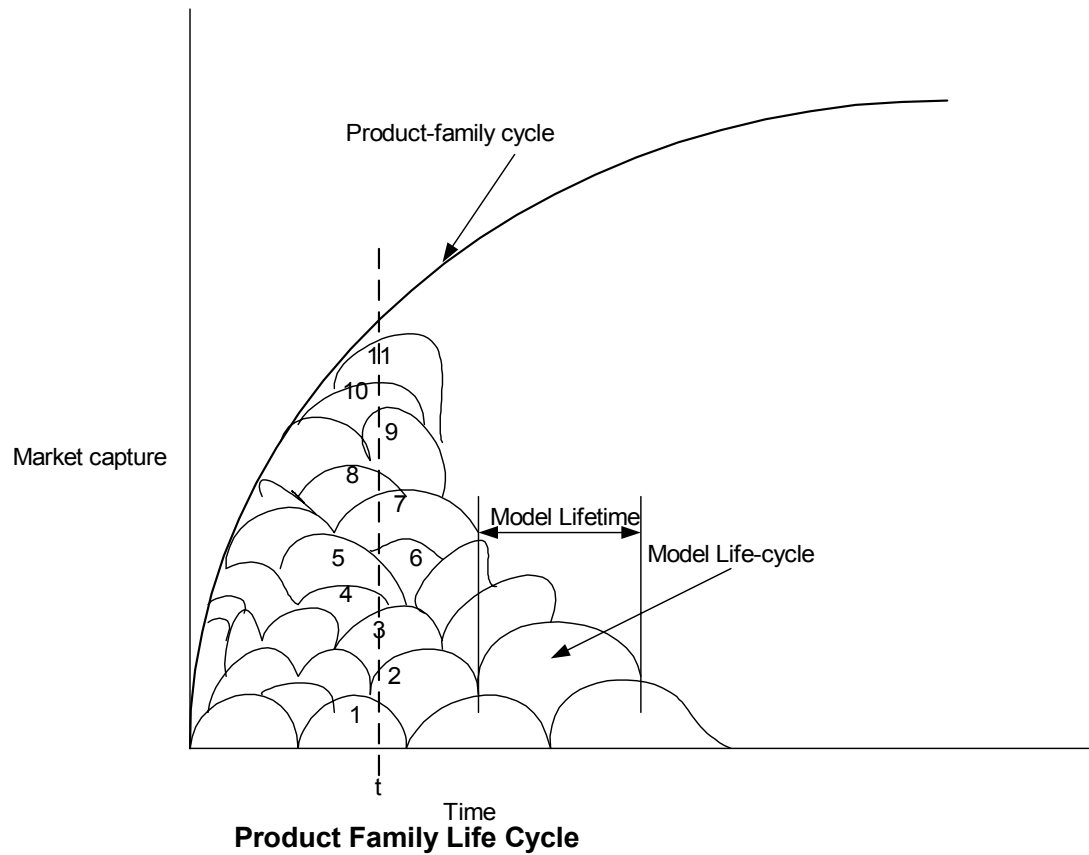
Figure 1.2: Product Family Life Cycle (Adapted from Uzumeri and Susan, 1997)

One must first clearly distinguish between the terms of model and family. Uzumeri and Susan define a *model* to be a product design that differs sufficiently from other designs that the manufacturer assigns it a distinctive commercial designation, and a *product* family to be a set of models that a given manufacturer makes and considers to be related (Uzumeri and Susan, 1997). Simpson defines *product family* as a group of products which share common *form features* and *function(s)*, targeting one or multiple market niches (Simpson, 1998). Here, *form features* refer generally to the shape and characterizing features of a product; *function* refers generally to the utilization intent of a product. A *derivative or product variant or model* is a specific instantiation of a product

7

platform within a product family which possesses unique form features and function(s) from other members in the product family. In Figure 1.2, it is seen that every model or product variant has its own lifetime. This is the time that the model remains in the market. After its lifetime, that model is retired and a new variant is brought into the market. The model life cycle represents the rise and fall of market capture of the model. It is seen that model 5 had a large market capture before half its lifetime. At its conception and end, its market share is very less.

In the early stages of the product family, less variety is provided, i.e., less models are present. As time passes, the variety increases to capture greater market share. After longer periods (not shown in figure), the market share of the products reduces. As more models that are targeted to distinct customer needs are developed, the market share of individual models tends to become smaller. So, it is not necessary that more variety will lead to larger market capture. However, the advantage is that increase of variety of the models leads to greater satisfaction of customers and hence major changes are not required in the models in the immediate future (Uzumeri and Susan, 1997). To have this design longevity, the already existing variety must preferably evolve to satisfy future needs of individual customers. Accordingly, the major challenges for mass customization are to provide such variety that can evolve as time passes. Competition and other factors make it absolutely necessary to extend the marketplace and capture other adjacent market spaces.

The above-discussed challenges of mass customization are also illustrated by considering the growth planning process of any company.

Table 1.2: Growth Planning Template

	Growth Planning Template: 2004-2014							
	Existing Customer Segments			**Future Customer Segments**				
Growth Target: $800M	Seg A	Seg B	Seg C	Seg D	Seg E	Seg F		
Existing Products	$20							
Replacement Products	$50							
New Products	$0							
Services	$10						**Total**	
Total Growth Objective	$80						$800	

Table 1.2 is a template that helps visualize the growth planning of any company X. It is seen that the company X has a growth target of $800M in the next 10 years. So how is it going to achieve this target? There are four types of sources of the company – existing products, replacement products, new products and services. Each of these sources has its one or more customer segments. The customer segments A, B and C are the existing markets, while customer segments D, E and F are the future markets. The total target from all these segments needs to come to $800M over the time period.

There are two problems that the company X faces.

1. The first problem that the company faces is how it should not only provide products in the existing customer segments but also plan for the future customer segments. The company may be able to achieve the targets in the existing customer segments by allocating resources for the short term; however, it is not able to achieve the targets of the future customer segments, due to lack of planning. Hence, at the end of 10 years, the growth target of $800M is not achieved.

    The important question to be answered is how can the company obtain a tradeoff between these two customer segments and achieve its 10 year goal. A part of this thesis addresses this question.

2.   Company X faces another problem. It has projected the targets of each customer segment based on some distribution of demand. Now, what if the demand in that particular customer segment changes? The target objectives will not be met. It is necessary that the targets be achieved despite of changes in demand in that segment. A part of the thesis addresses this problem.

If analyzed carefully, all the challenges discussed till now lead to the response that manufacturers must be able to extend the market lives of their product families. Design longevity plays a very important role in mass customization. It has been observed in history that there have been many such incremental design successes in which small changes have made product families to continue for years. Uzumeri and Susan (Uzumeri and Susan, 1997) have called such incremental design successes as "classics". Examples include the Kodak slide projector, Levi's jeans, the Coleman lantern, the grocery-store shopping cart, the Mason jar, the Harley-Davidson motorcycle, the Rolodex rotary card file and the original formula of Coca-Cola, rechristened Coke Classic. Any such long-lived design that reduces the level of product-line uncertainty represents a lifeline of stability in today's dynamic world. However, these 'classics' are very rare. Customer's needs and preferences change very rapidly. Manufacturers wish to offer families that match those needs. So the question is how can one design such long-lived designs in today's dynamic world? This question is addressed in the thesis by augmenting an existing method for product platform design to incorporate these needs.

In this section, we started with some of the basic challenges of mass customization. Today's key challenges of uncertain demand, heterogeneous niches and short product life cycles are identified and their importance analyzed in detail. At the end

of this section, the question of how should these challenges be tackled is put forward. Section 1.2 now deals with the existing product platform design methods in literature, their limitations and the requirements of the method for such a changing environment.

1.2 THE NEED FOR SYSTEMATIC PRODUCT PLATFORM DESIGN METHODS

1.2.1 Product Platform Design Methods in Literature

Manufacturing enterprises are recognizing that product design presents the best control over offering variety (Anderson, 1997). "Many companies are using platform-based product development to create product families that provide sufficient variety for the market while maintaining economies of scale and scope within their manufacturing processes" (Simpson, 2003). Product platforms are a set of common components, modules or parts from which a stream of derivative products can be created (Lehnerd, 1987). The reasoning behind platform design is to (1) simplify the product offering and reduce part variety by (2) standardizing components so as to (3) reduce manufacturing and inventory costs and (4) reduce manufacturing variability (i.e., the variety of parts that are produced in a given manufacturing facility) and thereby (5) improve quality and customer satisfaction (Simpson, 1998). The development of product platforms enables customization to the customer with minimum internal component variety. As such, platform design offers reduced development time and system complexity, reduced development and production costs, and improved ability to upgrade products (Simpson, 2003).

Tseng has corroborated this need: "it is imperative to develop a coherent framework within which systematic approaches can be taken to enable the realization of

mass customization" (Tseng et al., 1998). Developing systematic approaches for product platform design is therefore a pressing need and a relevant research problem.

Various engineering approaches have been proposed to develop product platforms and can be classified as either *bottom-up* or *top-down* (Simpson, 1999). Bottom-up approaches are the redesign and consolidation of existing products to create more competitive product families; thereby increasing economies of scale. Some of the different bottom-up approaches are modular based product family, group technology (Kalpakjian, 1997), product reasoning system (Siddique et al., 2000). For example, after working with individual customers to develop 100+ lighting control products, Lutron redesigned its product line around 15-20 standard components that can be configured into the same 100+ models from which customers could initially choose (Pessina & Renner, 1998). Black & Decker (Lehnerd, 1987) and John Deere (Shirley, 1990) have used similar redesign efforts to reduce variety in their motor and valve lines, respectively.

In top-down platform design, a company strategically manages and develops a family of products based on a common core and reduces redesign cost. The *Product Platform Concept Exploration Method* (Simpson et al., 2001) *Variation-Based Platform Design Methodology* (Nayak et al., 1999) fall in the category of top-down platform design. For example, Sony has strategically designed product platforms for the development of its Walkman® products (Sanderson & Uzumeri, 1997). Similarly, Kodak's product platform-based response to Fuji's introduction of the QuickSnap® single-use camera in 1987 enabled them to develop products faster and more cheaply. This made it possible to allow them to regain market share and eventually overtake Fuji (Wheelwright & Clark, 1995).

1.2.2 Limitations of Existing Platform Design Methods

Simpson has pointed out that irrespective of the top-down or bottom-up approaches, the methods are divided into two main categories of *Module-Based Product Family* and *Scale-Based Product Family* (Simpson, 2003). *Module-Based Product Family* is instantiated by adding, substituting, and/or removing one or more functional modules from the platform. *Scale-Based Product Family* uses one or more scaling variables to "stretch" or "shrink" the platform in one or more dimensions to provide variety. It has been noted that there are only a few platform design methods that can target both the categories. A summary of different features of methods in literature have been discussed in (Simpson, 2003) and is seen in Table 1.3. Almost two-thirds of the methods require specifying the platform *a priori* to the optimization in order to reduce the design space. This is not what is required in a good method since designers would like to use optimization to explore varying levels of platform commonality to help identify which variables to make common and unique within the family (cf., Simpson & D'Souza, 2002). Simpson has pointed out that more than half of the approaches use multi-objective optimization to accomplish this. However, more than half of the approaches do not model the demand and manufacturing costs. It has also been noted that the majority of approaches that include costs or sales in their formulation use single objective optimization, rather than multi-objective. Another important feature which is needed in today's uncertain environment is to consider changing extents of the marketplaces, as discussed in Section 1.1.3. The author has found only a few of the techniques tend to address this problem.

13

Table 1.3: Literature review of Product Platform Design methods (Adapted from Simpson, 2003)

Approach	Module-based family	Scale-based family	Specify platform a priori?	Single-objective	Multi-objective	Model manufacturing cost	Model market demand/sales	Consider uncertainty?
(Allada & Jiang, 2002)	x		Y	x			x	Y
(Blackenfelt, 2000b)	x		Y	x		x		Y
(Cetin & Saitou, 2003)	x		N		x			
(Chang & Ward, 1995)	x		Y	x				Y
(D'Souza & Simpson, 2003)		x	Y		x			
(Farrell & Simpson, 2003)		x	Y	x			x	
(Fellini, et al., 2000, 2002a, 2002b)	x	x	Y		x			
(Fujita, et al., 1998,1999)	x		Y	x		x	x	
(Fujita & Yoshida, 2001)	x	x	N	x		x	x	
(Gonzalez-Zugasti, et al., 2000, 2001)	x		Y	x	x	x	x	Y
(Hernandez,et al.,2001-03)(Carone,2002)(Williams,2003)	**x**	**x**	**N**	**x**	**x**	**x**	**x**	**-**
(Jiang & Allada, 2001)	x		N	x		x	x	Y
(Kokkolaras, et al., 2002)	x		Y		x			
(Li & Azarm, 2002)	x		Y	x	x	x	x	Y
(Messac, et al., 2002a, 2002b)		x	Y		x			
(Nayak, et al., 2002)		x	N		x			
(Nelson, et al., 2001)	x		Y		x			
(Ortega, et al., 1999)		x	N		x	x		
(Rai & Allada, 2002)	x		N		x	x	x	
(Seepersad, et al., 2000, 2002)		x	Y	x	x	x	x	Y
(Simpson, et al., 1999, 2001a, 2001b)		x	Y		x			

x represents that the method has a particular feature
- or blank space indicates the feature is absent

Apart from the lack of above mentioned features in existing platform design methods, there are two perceived major limitations of all top-down approaches listed. The first is related to the "extent" of commonality, the second, to the number of varied

specifications. The first limitation is concerned with top-down approaches used to commonalize features or components across an entire product family (i.e., the feature is made common to all the relevant product variants). This problem leads to a significant loss of performance, and has been discussed in the literature (e.g., (Nayak et al., 1999; Simpson et al., 1999; Seepersad et al., 2000; Nelson et al., 2001).

In order to reduce the impact of commonality on performance, one should be able to specify different levels of commonality for the various features and components of the product family. A method for determining platform extent that is based on the compromise Decision Support Problem (coupled with concepts derived from linear physical programming) has been listed in Table 1.3 (Seepersad et al., 2000). Unfortunately, this approach is only utilized with a limited number of products and does not consider *Module-Based Product Family* design. Designing products for mass customization implies a very large number of product variants for which this approach is impractical.

The second limitation is concerned with the number of varied specifications. Most of the product platform design methods have only one varied specification (e.g., the torque of a motor). Typically, products are customized for multiple specifications (e.g., the torque and the power of the motor) using multiple approaches for managing product variety (e.g., modular design, adjustable features, dimensional customization with CAD/CAM technology, etc.).

Thus, it is concluded that each product platform design technique suffers from one or more limitations. In response to these limitations, Hernandez proposes a novel top-down approach for developing product platforms that facilitates the realization of a

stream of customized product variants, and which accommodates the issue of multiple levels of commonality and multiple customizable specifications (Hernandez, 2001; Hernandez et al., 2002; Hernandez et al., 2003). The Product Platform Constructal Theory Method (PPCTM) formulates product platform design as a problem of access in a geometric space. Specifically, Hernandez shows that it is feasible and useful to formulate and solve systematically the design of hierarchic product platforms for customizable products as a problem of optimization of access in a geometric space (Hernandez et al., 2002). The result of the use of the PPCTM is a hierarchical organization of multiple approaches for offering variety, as well as the specification of their range of application across the product platform. Williams has been able to model demand and consider multi-objectives in PPCTM (Williams, 2003). So Hernandez and Williams together in their respective works have been able to address most of the mentioned characteristics in Figure 1.3 needed for a good method. The only characteristics missing in the method are that of consideration of uncertainty and changing extents of marketplaces. As discussed in Section 1.1.3, both of these are key challenges for mass customization.

Now that it is clear what is needed to make PPCTM alleviated of the major limitations, the requirements list is created (Table 1.4).

Table 1.4: Requirements list

Requirements List	
Problem Statement: *Augment the Product Platform Constructal Theory Method to deal with the uncertainty in distributions of demand, changing design parameters and extents of marketplaces?*	
Demand/ Wish	**Requirements**
	Specifics
D	Product platform robust to uncertain distribution of demand
D	Product platform robust to changing design parameters
D	Product platform adjustable to changing extents of marketplaces
	General
D	Must avoid costly process of redesigning product platforms
D	Must avoid costly process of designing replacements
W	Should obtain predictable sales
D	Give less strain on design resources
D	Free design resources for other innovation activities
W	Gives company time to realize reductions in design development costs and risks
W	Competitive flexibility
D	Simplify the management of product families

As discussed in Section 1.1.3, design longevity or the capability to have a design that can last longer, is one of the major challenges of mass customization and in effect the need to deal with uncertainty in distributions of demand, changing design parameters and extents of marketplaces. By making the product platform robust to such changes, firms get time to realize reductions in design development costs and risks. Less strain must be put on design resources. Suppose that a competitor launches a product in a new marketplace, the manufacturer should have the flexibility to extend his existing market space. The method should facilitate savings in redesigning. Suppose that the average

model life for a product family is 2 years. A design that will serve the market for 4 years will mean eliminating one complete redesign effort. This will free design resources for other activities like innovation. It is desirable to have predictable sales and if needed the design of replacements must not be costly. There should be a simplified way of managing the product families that will stabilize the evolution of the models. Thus, through this thesis, the author has augmented the existing PPCTM to develop a method that is not hampered by any of the above mentioned limitations.

1.3 RESEARCH QUESTIONS AND HYPOTHESES

The challenges for mass customization are presented in Section 1.1.3. The limitations of the existing product platform design approaches are discussed in Section 1.2.2. The key point coming out from the discussion in both the sections is the need of having a method that can:

- Handle uncertain distributions of demand

- Cope with changing design parameters

- Handle changes in the extents of marketplaces

In Section 1.2.2, it is observed that the Product Platform Constructal Theory Method covers most of the limitations of other methods except for the above mentioned requirements. Hence, the focus of the thesis is to augment PPCTM to alleviate it of these requirements.

There are two major thrusts in this thesis:

1. Augmentations to the PPCTM

- Infusion of concepts of robustness to deal with uncertain distribution of demand

- Enable the designer to consider changing design parameters

- Use of decision construct to consider objectives of initial as well as future possible extensions of marketplaces

2. The application of the PPCTM to two example problems

- The application of the PPCTM to the realm of product design for the development of a line of customizable pressure vessels

- The application of the PPCTM to the realm of product design for the development of a line of customizable hand exercisers

---

Primary Research Question

How should the Product Platform Constructal Theory Method be augmented to deal with the uncertainty in distributions of demand, changing design parameters and extents of marketplaces?

This primary research question is addressed by investigating a set of research questions and associated hypotheses.

Q1.1: *"How should the Product Platform Constructal Theory Method be augmented in order to cope with the uncertainty and inherent changes in distributions of demand?"*

Q1.2: *"How should the Product Platform Constructal Theory Method be augmented in order to cope with the changes in design parameters?"*

Hypothesis 1: The infusion of concepts of robustness into the Product Platform Constructal Theory Method enables to design platforms that are unaffected by variations in demand and design parameters.

Q2 *"How should the Product Platform Constructal Theory Method be augmented so that platforms are designed strategically considering future portfolio expansions?"*

Hypothesis 2: Compromise DSP can be used in the Product Platform Constructal Theory Method to make tradeoffs between objectives of the initial market extent and future probable extensions.

The result of this work is an augmented PPCTM that handles uncertain distribution of demand, changing design parameters and changing extents of marketplaces.

1.4 VALIDATION AND VERIFICATION STRATEGY FOR THIS THESIS

The validation and verification strategy used in this thesis is based on the validation square introduced by Pedersen and coauthors (Pedersen, Emblemsvag et al., 2000). As noted by Pedersen and coauthors, validation (justification of knowledge claims, in a modeling context) of engineering research has typically been anchored in

formal, rigorous, quantitative validation based on logical induction and/or deduction. This approach works fine for engineering design based primarily on mathematical modeling. Engineering design methods, however, rely on subjective statements as well as mathematical modeling; thus, validation solely by means of logical induction or deduction is problematic. Pedersen and coauthors propose an alternative framework in which "knowledge validation becomes a process of building confidence in its usefulness with respect to a purpose."

In this approach, the 'usefulness' of a design method is associated with whether the method provides design solutions correctly (structural validity) and whether it provides correct design solutions (performance validity). This process of validation is represented in the Validation Square in Figure 1.3.
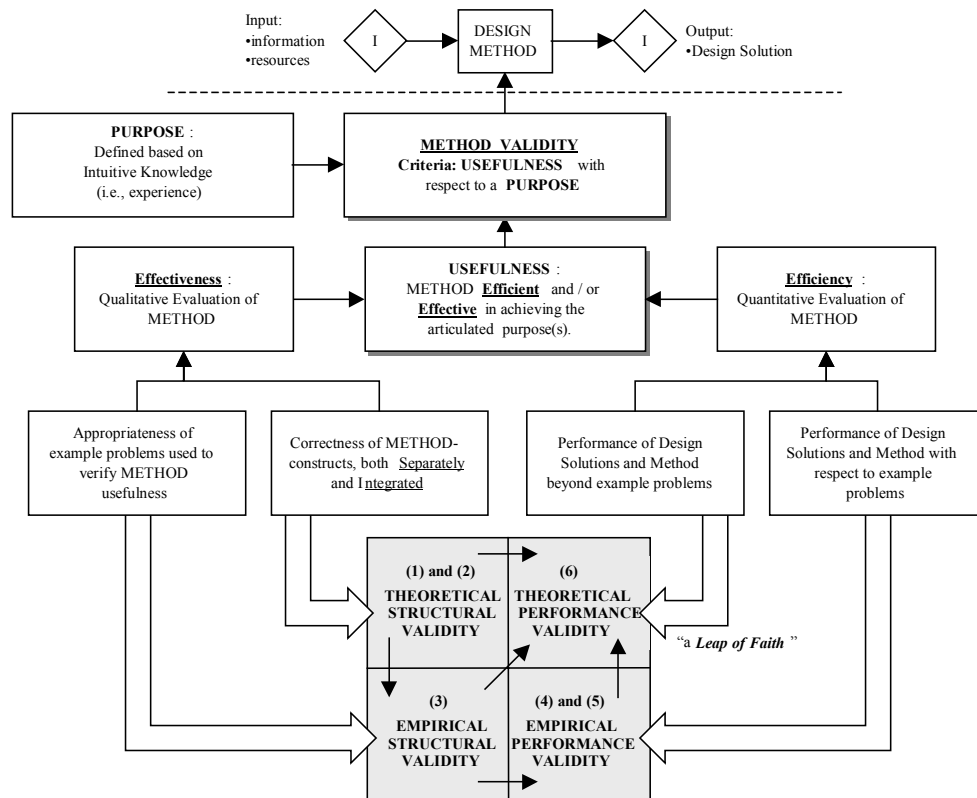
Figure 1.3: The Validation Square: Validating Design Theories or Methods (Pederson et al., 2000)

The specific manner in which the validation square is applied to this thesis is presented in Figure 1.4 and Figure 1.5. Inspired by the MS thesis of Seepersad (Seepersad, 2001), the Figure 1.5 is arranged according to the quadrants of the validation square, and references are included for chapters and sections in which the validation is documented.

Chapter 1 — Background

Chapter 2 — Theoretical Foundations

Chapter 3 — Method for Designing Product Platforms to deal with Uncertain Demands and Changing Marketplace

Chapter 6 — Closure

Theoretical Structural Validity

Theoretical Performance Validity

Empirical Structural Validity

Empirical Performance Validity

Chapter 4 / Chapter 5

Customized Product Platform Design in an Uncertain Environment: Pressure Vessels and Hand Exercisers		
Research Question	Section	Example
Q1.1	4.1	Pressure Vessel
Q1.2	4.2	Hand Exerciser

Customized Product Platform Design for a Changing Marketplace: Pressure Vessels		
Research Question	Section	Example
Q2	5.1, 5.2, 5.3, 5.4	Pressure Vessel

Figure 1.4: Chapters in the Validation Square

1.4.1 Theoretical Structural Validity

Theoretical structural validity involves searching and referencing the literature related to each of the constructs employed in the design method as well as the internal consistency of their assembly into the theory/method. In this thesis theoretical structural validity is presented in Chapters 2 and 3. This involves describing the PPCTM as well as the individual constructs in detail and critically reviewing each proposed augmentation. The analysis of the consistency of their combination and infusion into the PPCTM completes this quadrant of the validation square.

1.4.2 Empirical Structural Validity

Empirical structural validity involves building confidence in the appropriateness of the example problems chose for illustrating and verifying the performance of the design method. This includes documentation that the example problems chosen for validation are similar to the problems for which the methods/constructs are generally accepted. In other words, it must show that the example problems represent actual problems for which the method is intended, and that the data associated with the example problems can be used to support a conclusion.

Empirical structural validity is established in this thesis in Chapters 4 and 5. In these chapters two example problems are developed: the product platform design of a line of customizable pressure vessels, and the product platform design of a line of customizable hand exercisers.

1.4.3 Empirical Performance Validity

Empirical performance validity involves using representative example problems to evaluate the outcome of the design method in terms of its usefulness. Metrics for usefulness should be related to the degree to which the method's purpose has been achieved (e.g., reduced cost, reduced time, improved quality). It is also important to establish that the resulting usefulness is, in fact, a result of applying the theory/method.

In this thesis, empirical performance validity is also explored in Chapters 4 and 5. The usefulness of the augmentation is evaluated through the comparison of the results obtained from the use of the augmented PPCTM to results obtained without it.

1.4.4 Theoretical Performance Validity

Theoretical performance validity involves building confidence in the generality of the theory or method and accepting them as useful beyond the example problems. This includes showing that the problems are representative of a general class of problems and that the theory or method is useful for these problems. Thus, the general usefulness of the theory or method is inferred. The theoretical performance validity of the thesis is explored in the final chapter of this thesis.

1.5 ORGANIZATION OF THESIS

Theoretical Structural Validation is addressed in Chapter 2 and Chapter 3. Chapter 2 presents the theoretical foundation of the Product Platform Constructal Theory Method and its proposed augmentations. The Product Platform Constructal Theory Method is explained in detail to act as a foundation for other methods to augment it. The

theoretical foundations of Robust Concept Exploration Method and Compromise Decision Support Problem are then presented which are the bases of the augmentations proposed.

In Chapter 3, the infusion of the augmentations into the PPCTM is presented. Emphasis is placed on verifying the internal consistency of the method as a combination of the constructs presented in Chapter 2. Firstly, the method proposing to handle uncertain demand and design parameters is explained in detail. In the latter part, the use of Compromise DSP to design product platforms for a changing marketplace has been discussed. Finally, the resulting augmented PPCTM is presented.

Empirical structural and performance validity is offered in Chapters 4 and 5 through the presentation of two example problems. In the first section of Chapter 4, an example of the development of product platforms for customizable pressure vessels is discussed. The objective of using this example problem is to illustrate how one can design product platforms that handle uncertain demand. This is done using the method proposed in the first part of Chapter 3. The method is validated by showing the difference between product platforms designed without considering uncertain distributions of demand and platforms considering uncertain distributions of demand. The robust product platform designed is unaffected by changes in demand in the fixed market extent. This example is relevant since pressure vessel is a classic mechanical engineering problem which has been used a couple of times for validation of work of product platforms and it involves four modes of managing variety useful to illustrate this method.

In the second section of Chapter 4, an example of the development of product platforms for customizable hand exercisers is presented. The objective of using this

example problem is to illustrate how one can design product platforms that deal with changing design parameters. Similar concepts as in the first section of this chapter are used. The product platform is unaffected by changes in design parameters that may occur during the lifecycle of a platform. Hand exerciser is a product that often requires customization. It is a challenge for manufacturers to achieve such customization economically due to the competitive nature of the product. Therefore, this design problem is representative of a product that would benefit from the application of the theory and method proposed and is thus relevant.

In Chapter 5, the problem of designing a line of pressure vessels for a market that has a variable extent is discussed. The manufacturer of pressure vessels needs a method to strategically design product platforms that not only consider the objectives of the initial market space but also those of the future potential market expansions. Product platforms with a fixed extent will be useful for a manufacturer of pressure vessels for only a fixed amount of time. So it will not be possible to design the product platform with the demand of future customers in mind. Thus, the problem involves determining product platform specifications that gives preferred expected outcomes over time. The method is validated by considering two scenarios where in one case the platform is designed only for the current market space while in the other scenario the platform is strategically designed considering future potential expansions. It is seen that the strategically designed platform is much more effective in the longer run. The pressure vessel example helps to build on the method used to design a platform in the Chapter 4, where there was fixed market space. The author's primary objective in this chapter is to address the second section of the primary research question and hypothesis.

In the final Chapter 6, the research questions and the validity of the hypotheses are reviewed. The achievements and limitations in this thesis are discussed. Potential avenues of future work are investigated and closing remarks are made.

CHAPTER II

THEORETICAL FOUNDATIONS

With the background and motivation of the thesis discussed in the previous chapter, the theoretical foundations required for designing platforms and the associated augmentations are discussed. So the author's objective in this chapter is to present the Product Platform Constructal Theory Method and the fundamentals of the augments required to help answer the research questions presented in Section 1.3. After presentation of the PPCTM in Section 2.1, the theoretical foundations discussed are:

- Robust Concept Exploration Method
- Compromise DSP

Towards the end of this chapter, the concept of strategic design is discussed. Each of the theoretical foundations is critically reviewed. This chapter acts as the basis for the method to be developed in Chapter 3 that combines all these constructs.

		Relevance	**Hypotheses**

Foundation / Problem Identification

Chapter 1
Designing Product Families for Mass Customization

- Introduction, Motivation and Frame of Reference
- Research Questions and Hypotheses
- Organization of the thesis

Introduce

Chapter 2
The Theoretical Foundation

- Theoretical foundation of PPCTM
- Detailed description of Hernandez's PPCTM
- Description of constructs of potential augmentations
- Theoretical structural validation

Elaborate

Method

Chapter 3
A Method for Designing Product Platforms to Deal with Uncertain Demands, Changing Design Parameters and Changing Marketplaces

- Infuse concepts of robustness into product platform design
- Use compromise DSP for strategic product platform design
- Description of the method
- Theoretical structural validation

Method for Implementation of Hypotheses

Hypothesis Testing

Chapter 4
Customized Product Platform Design in an Uncertain Environment: Pressure Vessels and Hand Exercisers

- Demonstrate that the proposed augmentation is able to deal with changes in demand and design parameters.
- Empirical Structural and Performance Validity

Verify H1

Chapter 5
Customized Product Platform Design for a Changing Marketplace: Pressure Vessels

- Demonstrate that the use of compromise DSP is an effective way of strategically designing product platforms for a changing marketplace.
- Empirical Structural and Performance Validity

Verify H2

Closure

Chapter 6
A Look Back and A Look Ahead

- Summarize and critically review research findings
- Contributions and limitations
- Recommend future work
- Theoretical performance validation

Summarize

Figure 2.1: Thesis Roadmap

29

2.1 PRODUCT PLATFORM CONSTRUCTAL THEORY METHOD – PRODUCT PLATFORM DESIGN AS A PROBLEM OF ACCESS IN A GEOMETRIC SPACE

In this section the Product Platform Constructal Theory Method is presented. Sections 2.1.1 and 2.1.2 is comprised of the history, underlying concepts (hierarchic systems theory and constructal theory) and formulation of the PPCTM. After a detailed background of the PPCTM, the six-step method is presented in Section 2.1.3. Each step of the method is explained using the example of designing a line of customizable fans. The entirety of this section is based on the work presented by Hernandez in his doctoral dissertation (Hernandez, 2001) and is cited where appropriate.

2.1.1 Hierarchic Systems Theory

Simon has observed that complex structures adapt and evolve more efficiently when they are organized hierarchically (Simon, 1996). This is the crux of Hernandez's approach for organizing different modes of managing variety in a hierarchical manner. Specifically, Simon and Ando make two fundamental observations regarding an apparent tendency in the natural self-organization of complex systems (Simon et al., 1961):

1. Complexity, both in natural and artificial systems, frequently takes the form of a hierarchy, whereby *a hierarchic system is defined as being composed of interrelated subsystems that have in turn their own systems, and so on, until some elementary level of components is reached.*

2. In general, interactions inside subsystems (in a hierarchically-organized system) are stronger and/or more frequent than those interactions that occur between subsystems.

The concept of hierarchic systems is described as internal subsystems organized in ranks. One visualizes it better by considering aspects in the nature. Cells are considered as the building block, while higher in the hierarchy cell is organized into tissues, tissues into organs, organs into systems. Within the cell are well-defined subsystems: nucleus, cell membrane, microsomes and mitochondria.

A graphic example of a hierarchic system, and the counter example of a non-hierarchic system, is provided in Figure 2.2.
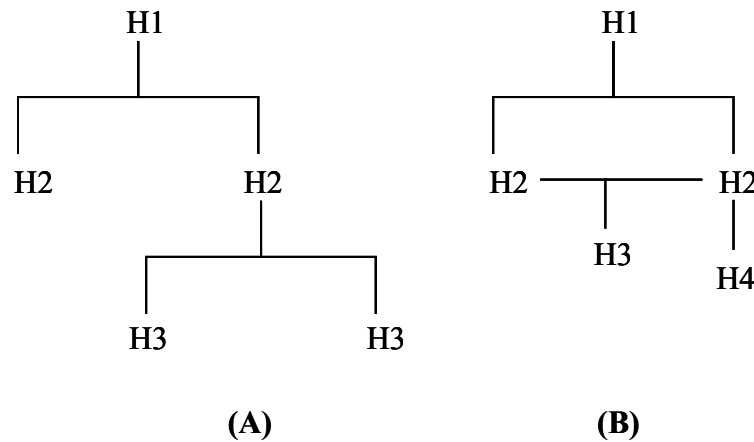
Figure 2.2: Hierarchic (A) vs. Non-Hierarchic (B) Organization of Systems (Williams, 2003)

Simon and Ando also present the concept of near-decomposability (Courtois, 1985). This deals with the concept that favorable conditions arise when each stable subsystem operates nearly independently of the processes happening within the other subsystems.

The concepts of hierarchic organization and near-decomposability are two fundamental posits of the PPCTM (Hernandez, 2001):

Posit 1: Potential for rapid adaptation and/or response is higher in complex systems when they are organized hierarchically.

Posit 2: In hierarchically organized systems, the high-frequency (short run) responses tend to be associated with the lowest levels of hierarchy and the low-frequency (long run) ones with the interactions of these subsystems, i.e., the higher levels of the hierarchic organization.

With the idea of organizing the modes of managing product customization as a hierarchy, Hernandez searched for a construct in which the modes could be efficaciously organized.

2.1.2   Constructal Theory

In Section 2.1.1, the use of concepts of hierarchical system theory that facilitate complex systems to adapt to changes in the environment of different magnitude and frequency are done (Posits 1 and 2). In this sub-section, along with the Posits of the previous section, a series of postulates from constructal theory (Bejan, 2000) are identified that constitute the theoretic basis of a theory and method for designing product platforms.

Constructal theory, initiated by Adrain Bejan in 1996, is a result of studying optimal access in a number of flow and traffic problems (Bejan, 1996). The principle behind this theory is the constrained optimization of global performance – as the generating mechanism of organization, complexity and hierarchic structure in nature, engineering and even management. The concept is that hierarchic organizations observed in nature are the result of a sequential optimization process with the objective of the maximization of access (or minimization of resistance, or losses). This optimization process should proceed in a specific time direction: from the optimization of the basic

elements to the higher-order assemblies of the structure. For example, consider the formation of tree networks observed in nature such as blood vessels, tree branches, river tributaries, etc. All of these are considered as a hierarchic process of optimization: the shape that optimizes "access" at the most elementary volume occurs first, followed by an assembly of these innermost spaces into a second, larger, shape, which optimizes "access" at that level. These second shapes in turn are assembled into a third space, and so on. This sequential process continues until each space is connected and the entire space is covered, as shown in Figure 2.3.
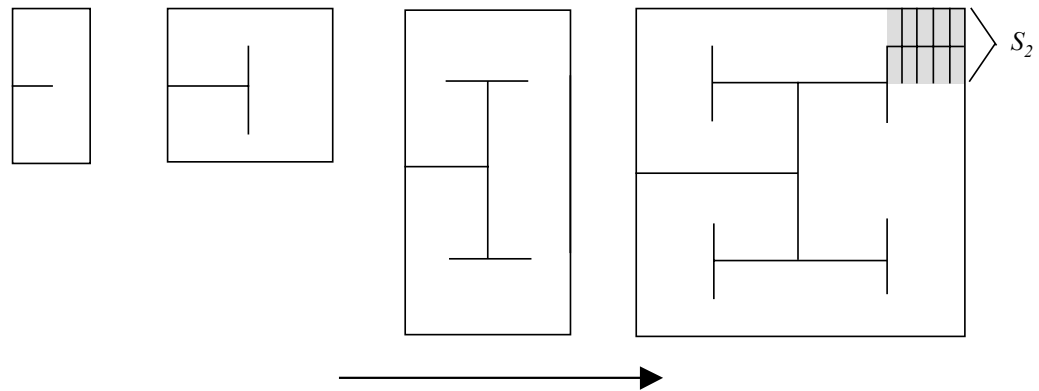
Figure 2.3: Organization in Nature as a Problem of Access Optimization (Bejan, 2000)

Hernandez uses a similar approach to organize modes of managing product customization. The modes are arranged in a hierarchic manner based on the representation of platform design as a problem of optimization of access in a geometric space. An optimal access problem is characterized by the need to determine the optimal "bouquet of paths" that links all points of an area S with a common destination O (Figure 2.4).
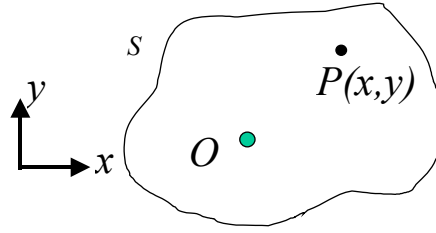
Figure 2.4: A Finite-Size Area with a Common Destination (Bejan, 1997)

The arrangement of these paths is usually optimized towards some objective (i.e., most efficient, least resistance, minimal travel time). The "access problem" and the constructal solution are better understood via a street-network example taken from (Bejan, 1996; Bejan, 1997; and Bejan, 2000). Consider a finite-size geographical area S and a point O situated inside S or on its boundary (Figure2.5). Each member of the population living on S must travel between a point of residence $P(x,y)$ and the point O. The problem is to connect every point in the area S to this point O. The obvious solution would be to have a straight-lined street to every point from O. However, considering existence of faster modes of transportation (say at a velocity by an automobile $V_1 > V_o$ via walking) there is a limitation on the areas on which some modes cannot travel. As such, the solution then is to bring a finite number of streets near patches of land where inhabitants can access the streets (in order to travel at a velocity, V_1) via walking (V_o).
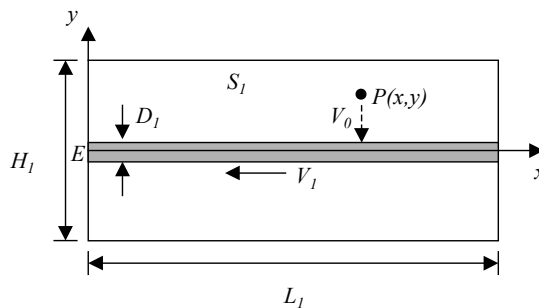
Figure 2.5: The First Area Element (Bejan, 1997)

34

Fundamentally, the access problem of this example is allocating a finite length of street to each finite patch of area, and also connecting these street lengths in an optimum manner such that travel time is minimized.
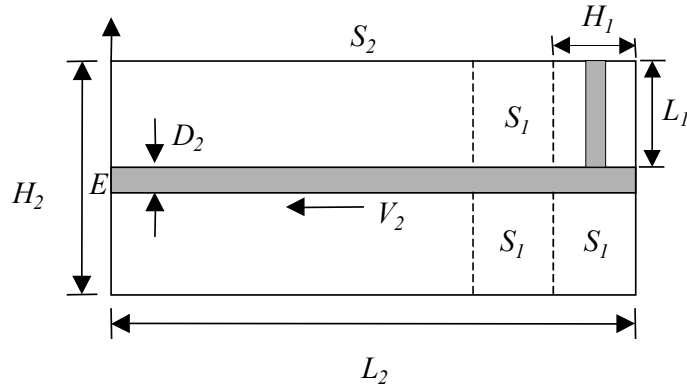
Figure 2.6: The Second Area Element (Bejan, 1997)

The direction of the solution by constructal theory is to assemble geometrically optimized areas (from smallest to largest) until the entire area, *S*, is covered as shown schematically in Figure 2.6.

It has been shown, however, that this sequential optimization process yields sub-optimal results (Hernandez, 2001). Bejan's constructal theory, however, provides the key to formulate problems of access in geometric spaces as multi-stage optimization problems for which more effective solution algorithms can be utilized.

The outcome of Hernandez's research of constructal theory was three additional posits to the PPCTM (Hernandez, 2001):

> *Posit 3:* Systems complexity results from a natural process of systems to provide paths of easier access.

Posit 4: Each path of access within the optimized system structure is unique and does not cross with other paths: the resulting structure is hierarchic.

Posit 5: The design of a hierarchic structure to provide easier access should proceed in a specific time direction: from the optimization of the basic elements at the smallest scale towards the optimal arrangement of these elements into higher-order assemblies, the process being one of repeated maximization of access (or minimization of losses) subject to constraints.

With its theoretic foundation established, the Product Platform Constructal Theory Method itself is presented next.

2.1.3   Steps in the Product Platform Constructal Theory Method

In this section, the six steps of Hernandez's PPCTM are discussed. The steps are explained using the examples of customizable fans and cantilever beams, similar to (Hernandez, 2001) and (Williams, 2003) respectively. Two examples are used for a couple of reasons. Firstly, fans and beams have one and two dimensional market spaces respectively, similarly, hand exercisers and pressure vessels also have one and two dimensional market spaces respectively. Hence it is easier to understand the examples in Chapters 4 and 5. Secondly, fans help to give a higher-level picture of development of platforms using the PPCTM while beams help to go into details of the method. Moreover, it helps to highlight the different limitations of the existing PPCTM by considering two different examples.

```
┌─────────────────────────────────────────┐
│      Step 1. Define the Market Space      │
└─────────────────────────────────────────┘

┌─────────────────────────────────────────┐
│   Step 2. Formulate an Objective Function │
└─────────────────────────────────────────┘

┌─────────────────────────────────────────┐
│      Step 3. Identify Modes for Managing  │
│              Product Variety              │
└─────────────────────────────────────────┘

┌─────────────────────────────────────────┐
│    Step 4. Identify Number of Stages and  │
│  Define a Baseline Decision for Each Stage│
└─────────────────────────────────────────┘

┌─────────────────────────────────────────┐
│   Step 5. Formulate the Multistage Decision│
│        Problem as a Dynamic Program        │
└─────────────────────────────────────────┘

┌─────────────────────────────────────────┐
│      Step 6. Solve the Dynamic Program    │
└─────────────────────────────────────────┘
```
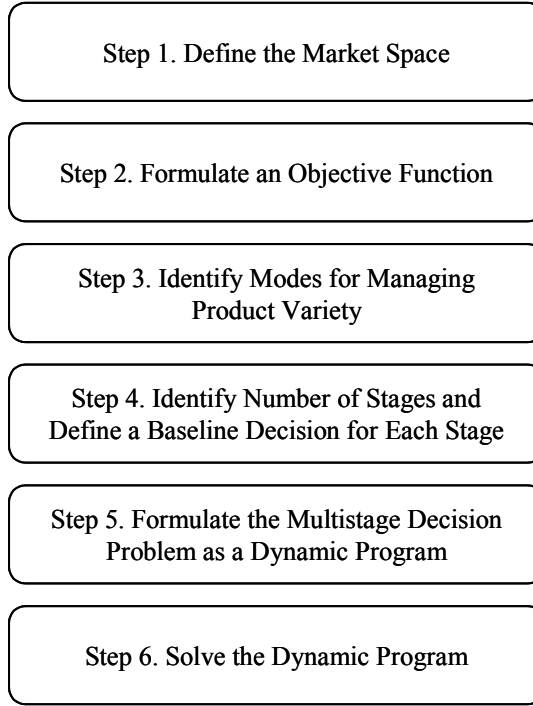
Figure 2.7: The Six Steps of the Product Platform Constructal Theory Method (Hernandez, 2001)

Step 1: Define the Market Space

In the first step of the PPCTM the space of customization is defined. The *space of customization* is the set of all feasible combinations of values of product specifications that a manufacturing enterprise is willing to satisfy (Hernandez et al., 2002). Consider that there are N independent product requirements $r_1, r_2,...,r_N$ identified that characterize the customer demands on a product. These requirements help to define the N-dimensional space of customization $M_n = \{(r_1, r_2, ..., r_N)\}$.

The number of product specifications that will experience variety determines the dimension of the space. The ranges of the specifications for which variety will be offered define the bounds of the geometric space to be analyzed. This range is usually bounded by economic or technological limitations.

Example 1: Consider FanTech, a company that manufactures fans and satisfies customer needs by providing customizable products. The logic behind providing such a variety is that it will create a product of differentiation for the company and thus a competitive advantage. The embodiment design of their fan is presented in Figure 2.8.

(a) Basic Components of an Existing Fan

Speed Controller

Smaller Motor

Smaller Blades Diameter

(b) Three Options for Varying the Capacity of the Fan
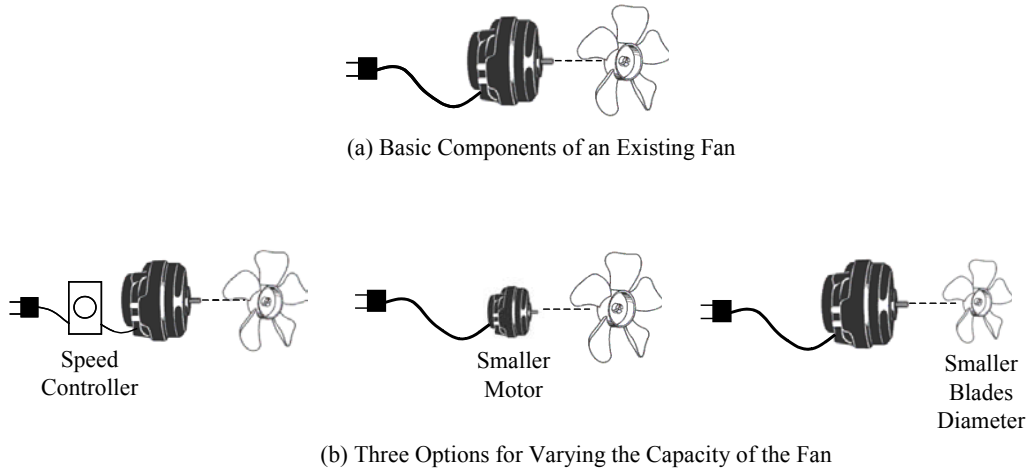
Figure 2.8: Three Possible Options for a Derived Product Realization (Hernandez, 2001)

The defining requirement of the customers of FanTech is the airflow capacity. Market research carried out by FanTech shows that there is a demand for the capacity in between 20 and 500 L/s. Thus, the market space consists of a continuous set of one requirement, i.e., a one-dimensional market space $M^1 = \{r_1\}$ with $20 \leq r_1 \leq 500$. The fans require at least three components (the technology base): a motor, a set of blades and an electric cord (Figure 2.8).

Example 2: Consider another company BeamTech that provides a line of customizable cantilever. The embodiment design of their cantilever beam is presented in Figure 2.9.

Figure 2.9: Customizable Cantilever Beam (Williams, 2003)

The defining requirements of BeamTech are the length (inches) and the maximum applied load it can withstand before yielding (lbs). The range of length demanded is from 100 to 200 inches and the range of maximum end load demanded is from 1000 to 5000 lbs. Thus, the space of customization consists of a two-dimensional market space $M^2 = \{r_1, r_2\}$ with $1000 \leq r_1 \leq 5000$ and $100 \leq r_2 \leq 200$. The design parameters involved are:

- E: the modulus of elasticity (psi); a material property

- σ_u: ultimate strength (psi); a material property

- I: moment of inertia (in^4); a property of the cross-section

- c: largest distance from the neutral surface (in.); a property of the cross-section
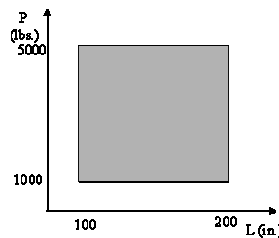
Figure 2.10: Space of Customization for Cantilever Beam Example

Step 2: Define the Objective Function

The second step of the method involves identifying the objective of this product platform to be designed. Some examples of such objective functions for product families

include the minimization of average cost, or the maximization of the profit; any other suitable design parameter such as quality or performance is acceptable.

Example 1: Considering again the example of the fan manufacturer, FanTech wishes to minimize the average cost of the entire product family. There could be any other objective possible related to the efficiency of the system, minimizing the weight of the fan, etc. The cost of a fan is assumed to be function of the motor, a set of blades and an electric cord *C(M, B, E)*.

The cost of the product platform for fans in the entire market space needs to be first defined in terms of the variable product parameter defined in Step 1 (air flow capacity). Then it is integrated over the market space:

$$c = \int_{a\min}^{a\max} C(a)da \qquad [2.1]$$

If the example were taken for a market space of discrete values, the objective function would simply be calculated as the sum of the cost of all the product variants in the entire space:

$$c = \sum_{a\min}^{a\max} C_i(a) \qquad [2.2]$$

For the PPCTM's working assumption of uniform demand, the average cost is simply evaluated as the quotient of Equation 2.1 and the demand, *D*.

$$c = \frac{1}{D} \int_{a\min}^{a\max} C(a)da \qquad [2.3]$$

40

Example 2: Considering again the example of the beam manufacturer, BeamTech wishes to minimize the average cost of the entire product family. The cost of a beam is assumed to be function of length, neutral surface distance, moment of inertia, and ultimate strength, $C(L, c, I, \sigma_u)$.

The cost of the product platform for fans in the entire market space needs to be first defined in terms of the variable product parameters defined in Step 1 (length of beam and load). Then it is integrated over the market space:

$$c = \int_{L_{min}}^{L_{max}} \int_{P_{min}}^{P_{max}} C(P,L)dPdL \qquad [2.4]$$

If the preceding example were taken for a market space of discrete values, the objective function would simply be calculated as the sum of the cost of all the product variants in the entire space:

$$c = \sum_{j=P_{min}}^{P_{max}} \sum_{i=L_{min}}^{L_{max}} C_{i,j}(P,L) \qquad [2.5]$$

For the PPCTM's working assumption of uniform demand, the average cost is simply evaluated as the quotient of Equation 2.1 and the demand, D.

$$c = \frac{1}{D} \int_{L_{min}}^{L_{max}} \int_{P_{min}}^{P_{max}} C(P,L)dPdL \qquad [2.6]$$

Step 3: Identify the Modes for Managing Product Variety

In the step, the question of how to vary a product concept in order to satisfy all the customized specifications of the market space is addressed. The modes of managing variety represent the ways in which one can provide variety to the customer in the market space. These are the linking mechanism between the product variants that compose the product family. Some modes for managing product variety include dimensional

commonalization, component commonalization, modularity, and standardization. There is no defined way of carrying out this step. This step instead involves brainstorming and decision-making between design and manufacturing.

Example 1: In the example of FanTech, the manufacturer considers the following three modes for managing product variety:

Standardization of motors – This involves use of different standardized motors.

Standardization of blades – The use of standardized sizes of blades helps to minimize costs.

Customization of air flow capacity using a speed control – The use of standard speed controllers helps to control the air flow capacity.

Example 2: In the example of BeamTech, the manufacturer considers the following three modes for managing product variety:

1) *Customization of beam length*: The manufacturer decides to order raw lengths of the beam material from which the customer's desired length is cut-to-order. While this minimizes the cost associated with ordering, it will add a cost associated with waste. This mode offers variety along the axis of length.

2) *Standardization of cross-section*: This involves use of standardized cross-section of the beam is standardized. This mode offers variety along the axis of maximum applied load.

3) *Standardization of material*: The use of standardization of material offering variety along the axis of maximum applied load minimizes cost through discounts in ordering.

Step 4: Identify Number of Stages and Define a Baseline Decision for Each Stage

Now that the space of customization and the objectives of the product platform are defined, one needs to identify the modes for managing product variety. This step involves the consideration of how a product will be varied in order to satisfy the customized specifications of the space of customization. The number of hierarchy levels can be equal to or less than the number of modes of managing variety. A hierarchy level can have one or more modes of managing variety. These modes of managing product variety are the linking mechanism between the product variants that compose the product family. There is no rigid structure to the formulation of these modes; this step instead involves brainstorming and decision-making between design and manufacturing.

From Hernandez's second posit (Section 2.1.1) it is established that the modes for managing product customization capable of achieving the smallest variations in design parameters should be used first. Some approaches, such as dimensional customization, can be used to satisfy very small variations in the product, but are normally expensive for large changes. Other approaches, such as modularity, are not economically feasible for small changes. As such, economical and technological considerations play an important role in establishing the hierarchic use of the modes for managing product variety.

After the establishment of order of the use of the modes for managing product variety, the baseline decision to be made at each stage must be formulated. This decision is simply the selection of the design parameters that minimize (or maximize) the given objective function for that specific stage.

The decision variables for a stage define the sub-space on which one or more modes for managing product variety are utilized. For a problem with *N* product requirements, the decision-variables for any stage *i* are

$$\Delta r(i) = [\Delta r_1(i), \Delta r_2(i), ..., \Delta r_N(i)] \qquad (2.7)$$

In the formulation of the baseline decision for each stage it is required, in order to achieve a hierarchic design (Hypothesis 2.1), that

$$\Delta r_j(i+1) \geq \Delta r_j(i) \qquad (2.8)$$

A generic template for formulating the baseline decision is shown in Figure 2.11.

For Stage *i*

Given The N-dimensional market space $M^N = \{(r_1, r_2, ..., r_N)\}$

     The decision variables of previous stages $\Delta r(1), ..., \Delta r(i-1)$

     The modes for managing product variety to be utilized at Stage i

     The set of constraint functions $g[\Delta r(1), ..., \Delta r(i-1)] \geq 0$

     An objective function $f_i[\Delta r(1), ..., \Delta r(i)]$

Find   The value of the decision variables $x(i) = [\Delta r_1(i), \Delta r_2(i), ..., \Delta r_N(i)]$

Satisfy *Constraints* $\Delta r_j(i) \geq \Delta r_j(i-1)$

     Bounds $\Delta r_{j\min} \leq \Delta r_j(i) \leq \Delta r_{j\max}$

Optimize The value of the objective function for that stage

     $f_i = f_i[\Delta r(1), ..., \Delta r(i)]$

Figure 2.11: Baseline Decision for an Arbitrary Stage *i* (Hernandez, 2001)

There are constraints on the decision variables such that the increment in each design variable is not greater than the corresponding increment for the previous stage. Bounds exist in the decision to prevent the decision variable increment from being smaller than itself, or larger than the amount of variability desired.

Example 1: In the case of the fan example, it is observed that trying to fulfill any possible capacity using a particular combination of motors and blades requires a very large number of variants of either motors or blades, which is not practical or economical. The only mode that can be utilized to achieve efficaciously very small variations is the third mode, i.e., the use of a speed control. Using the Equation 2.3, with the objective to minimize it, one obtains the appropriate speed control values keeping other values constant. Fundamentally, the crux of this decision is the identification of the range for which the speed controller will be commonalized (ΔA_1).

> **For Stage 1**
>
> **Find**   The value of the decision variable ΔA_1
>
> **Satisfy**   *Bounds* $\Delta A_1 \leq A_{max}$
>
> **Optimize**   The value of the objective function for this stage:
>
> $$c_1 = \frac{1}{D} \int\limits_{a\,\min}^{a\,\max} C(a_1)\,da$$

Figure 2.12: Baseline Decision for Stage 1 of Fan Example

Now, for the second stage, consider that it is easier to provide variety in motors than variety in blades. Hence, the second mode for managing product variety, standardization of motors, is now utilized for the second space element and the problem is how many of the space elements previously defined should be "connected" with a single

motor. Again, the Equation 2.3 is minimized and the optimal number is obtained.

For Stage 2

Given   The value of ΔA_1

Find   The value of the decision variable ΔA_2

Satisfy   *Bounds* $\Delta A_1 \leq \Delta A_2 \leq A_{max}$

Optimize   The value of the objective function for this stage:

$$c_2 = \frac{1}{D} \int_{a\min}^{a\max} C(a_2)da$$

Figure 2.13: Baseline Decision for Stage 2 of Fan Example

Fundamentally, the crux of this decision is the identification of the range for which the motor will be commonalized (ΔA_2).

For Stage 3

Given   The value of ΔA_2, ΔA_1

Find   The value of the decision variable ΔA_3

Satisfy   *Bounds* $\Delta A_2 \leq \Delta A_3 \leq A_{max}$

Optimize   The value of the objective function for this stage:

$$c_3 = \frac{1}{D} \int_{a\min}^{a\max} C(a_3)da$$

Figure 2.14: Baseline Decision for Stage 3 of Fan Example

Fundamentally, the crux of this decision is the identification of the range for which the blades will be commonalized (ΔA_3). The result of this process is a *hierarchic* organization of the modes for managing product variety that maps *any* airflow capacity (the magnitude of the function specification) with a set of components (the product platform).

Example 2: In this step for the cantilever beam example, the previously proposed modes of managing customization need to be placed in a hierarchy. For the first stage, the mode that offers the maximum variability in the most cost effective manner is the mode of customization of beam length (Mode L). Fundamentally, the crux of this decision is the identification of the range for which the raw beam length will be commonalized (ΔL).

For Stage 1

Find The value of the decision variable ΔL

Satisfy *Bounds* $\Delta L \leq L_{max}$

Optimize The value of the objective function for this stage:

$$c_1 = \frac{1}{D} \int\limits_{L_{min}}^{L_{max}} \int\limits_{P_{min}}^{P_{max}} C(P, L_1)\, dP\, dL$$

Figure 2.15: Baseline Decision for Stage 1 of Cantilever Beam Example

Since there are two parameters in which variety needs to be offered, the second stage involves the mode that provides most variability in the load. Standardization of Cross-Section (Mode P_1) provides such variety.

For Stage 2

Given The value of ΔL

Find The value of the decision variable ΔP_1

Satisfy *Bounds* $\Delta P_1 \leq P_{max}$

Optimize The value of the objective function for this stage:

$$c_2 = \frac{1}{D} \int\limits_{L_{min}}^{L_{max}} \int\limits_{P_{min}}^{P_{max}} C(P_1, L)\, dP_1\, dL$$

Figure 2.16: Baseline Decision for Stage 2 of Cantilever Beam Example

Fundamentally, the crux of this decision is the identification of the range for which the cross-section of the beam will be commonalized (ΔP_1).

Standardization of Material (Mode P_2) is the third and last stage since it offers the smallest amount of variability and at the most expense.

For Stage 3

Given   The value of ΔL, ΔP_1

Find   The value of the decision variable ΔP_2

Satisfy   *Bounds* $\Delta P_1 \leq \Delta P_2 \leq P_{max}$

Optimize   The value of the objective function for this stage:

$$c_3 = \frac{1}{D} \int_{L_{min}}^{L_{max}} \int_{P_{min}}^{P_{max}} C(P_2, L) dP_2 dL$$

Figure 2.17: Baseline Decision for Stage 3 of Cantilever Beam Example

Fundamentally, the crux of this decision is the identification of the range of the commonality of the material choice (ΔP_2).

Step 5: Formulate the Multistage Decision Problem as a Dynamic Program

The baseline decisions of each stage established in the previous step are formulated as a multistage optimization problem in the implementation of the PPCTM. The fifth step of the PPCTM suggests the use of dynamic programming to solve this complex problem. Dynamic programming (Jacobs, 1967) is an approach for optimizing multistage decision processes and is based on Bellman's principle of optimality:

> *Principle of Optimality*: An optimal policy in a multi-stage optimization process has the property that regardless of the decisions taken to enter a particular stage, the remaining decisions must constitute an optimal policy for leaving that state.

Dynamic programming begins with the last stage of an *n*-stage process and determines for each state the best policy for leaving the stage and completing the process, assuming that all preceding stages have been completed. The decision maker then moves "backwards" through the process stage by stage.

Formulating the multi-stage problem as a dynamic program involves:

o   Defining the optimal objective function for each stage

o   Formulating the general recurrence equation.

o   Developing a table or closed-form equation for the optimal objective function for each stage

o   Formulating the decision to be made at each stage of the solution process

Dynamic programming is based on the concept that the optimal parameters for the global problem are based on the selection of the parameters through the knowledge of the optimal parameters at each stage of the decision problem.   The formulation of the multistage problem as a dynamic program is shown in Figure 2.18.

For Each Stage *i*
Given   The N-dimensional market space $M^N = \{(r_1, r_2, ..., r_N)\}$
The decision variables of previous stages $\Delta r(1), ..., \Delta r\ (i\text{-}1)$ The modes for managing product variety to be utilized at Stage i   The set of constraint functions $g[\Delta r(1), ..., \Delta r\ (i\text{-}1)] \geq 0$ An objective function $f_i\ [\Delta r(1), ..., \Delta r\ (i)]$
Find   The value of the decision variables $x(i) = [\Delta r_1(i), \Delta r_2(i), ..., \Delta r_N(i)]$
Satisfy   *Constraints* $\Delta r_j(i) \geq \Delta r_j(i-1)$
Bounds $\Delta r_{j\min} \leq \Delta r_j(i) \leq \Delta r_{j\max}$
Optimize   The value of the objective function for that stage $$f_i^* = f_i\left[\Delta r(1), ..., \Delta r(i)\right] + f_{i+1,k}^*\left[\Delta r(1), ..., \Delta r(i)\right]$$
where the variable f_i^* is the new objective function for any stage *i*.

Figure 2.18: Formulation of the Multi-Stage Problem as a Dynamic Program (Hernandez, 2001)

Example 1: There are three decision stages in the fan example which are formulated into an overall multi-stage formulation:

Find The value of the decision variable ΔE, ΔM, ΔB

Satisfy *Bounds* $\Delta E \leq \Delta E_{max}$

$\Delta M \leq \Delta M_{max}$

$\Delta B \leq \Delta B_{max}$

Optimize The objective function (same as the third stage):

$$c_3 = \frac{1}{D} \int_{a\,min}^{a\,max} C(a_3)da$$

Figure 2.19: The Multistage Problem for the Fan Example

Example 2: There are three decision stages in the cantilever beam example which are formulated into an overall multi-stage formulation:

Find The value of the decision variable ΔL, ΔP$_1$, ΔP$_2$
Satisfy *Bounds* $\Delta L \leq L_{max}$
$\Delta P_1 \leq P_{max}$
$\Delta P_1 \leq \Delta P_2 \leq P_{max}$
Optimize The objective function (same as the third stage):
$$c_3 = \frac{1}{D} \int_{L_{min}}^{L_{max}} \int_{P_{min}}^{P_{max}} C(P_2, L)dP_2dL$$

Figure 2.20: The Multistage Problem for the Cantilever Beam Example

Step 6: Solve the Dynamic Program

In the final step of the PPCTM, the dynamic program established (previous step) is solved. The solution involves the development of response surface in order to

approximate optimal objective functions for each stage by moving "backwards" through

the problem and then in a forward manner through the use of these response surfaces.
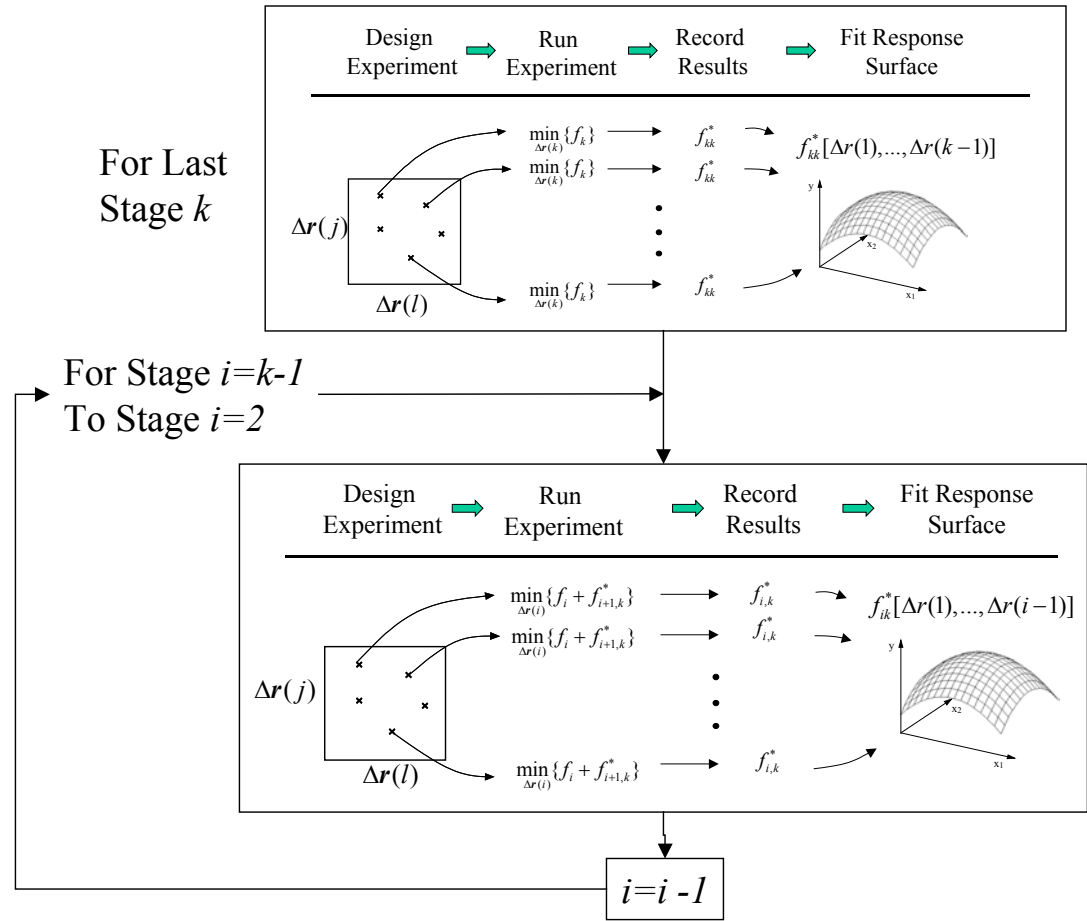
Figure 2.21: Developing Response Surface Approximations for Optimal Objective Functions in Dynamic Programming (Hernandez, 2001)

As seen in Figure 2.21, the process for approximating optimal objective functions

using the response surface method involves a multi-step process (Hernandez, 2001):

1. For the last stage k of the process:

    a. An experiment is designed with the value of the decision variables from

    previous stages, $\Delta r(1)$, $\Delta r(2)$, …, $\Delta r(k\text{-}1)$ as independent factors and the value of

    the optimal objective function, f_k as the response variable.

b.   For each run (i.e., each combination of the decision variables $\Delta r(1)$, $\Delta r(2)$, … $\Delta r(k\text{-}1)$) from the experimental design, the value of f_k is optimized by choosing the appropriate value of $\Delta r(k)$ and the value of the optimized function obtained, referred to as f_{kk}^*, is recorded.

c.   After all runs are completed, a response surface is fit to the optimized values of f_{kk}^* (recorded from the previous step) as a function of the decision variables of previous stages $\Delta r(1)$, $\Delta r(2)$, … $\Delta r(k\text{-}1)$. This response surface is an approximation of the optimal objective function for Stage k.

2.   For Stage $i = k\text{-}1$ to Stage $i = 2$:

a.   An experiment is designed with the value of the decision variables from previous stages, $\Delta r(1)$, $\Delta r(2)$, … $\Delta r(i\text{-}1)$ as independent factors and the value of $f_i + f_{i+1,k}^*$ as the response variable, where the optimal objective function, $f_i + f_{i+1,k}^*$, has been obtained in the preceding step.

b.   For each run (i.e., each combination of the decision variables $\Delta r(1)$, $\Delta r(2)$, …, $\Delta r(i\text{-}1)$) from the experimental design, the value of $f_i + f_{i+1,k}^*$ is optimized by choosing the appropriate value of $\Delta r(i)$ and the value of $f_i + f_{i+1,k}^*$ obtained is recorded. This value is referred to as $f_{i,k}^*$.

c.   After all runs are completed, a response surface is fit to the values of $f_{i,k}^*$ recorded from the previous step as a function of the decision variables of previous stages $\Delta r(1)$, $\Delta r(2)$, …, $\Delta r(i\text{-}1)$. This response surface is an approximation of the optimal objective function for Stage $i\text{-}1$.

d. $i = i - 1$ and steps (a) through (c) are repeated.

The solution of this dynamic program results in the selection of optimal ranges of application for each mode of managing product variety. These values are then used to map the design parameters to a fixed set of components that can be varied using the modes of managing variety in order to meet individual customer specifications.

Example 1: An exhaustive search of the different combinations of the decision variables is sufficient due to the simplicity of the examples. The three-loop iteration of the variables is shown in Figure 2.22.
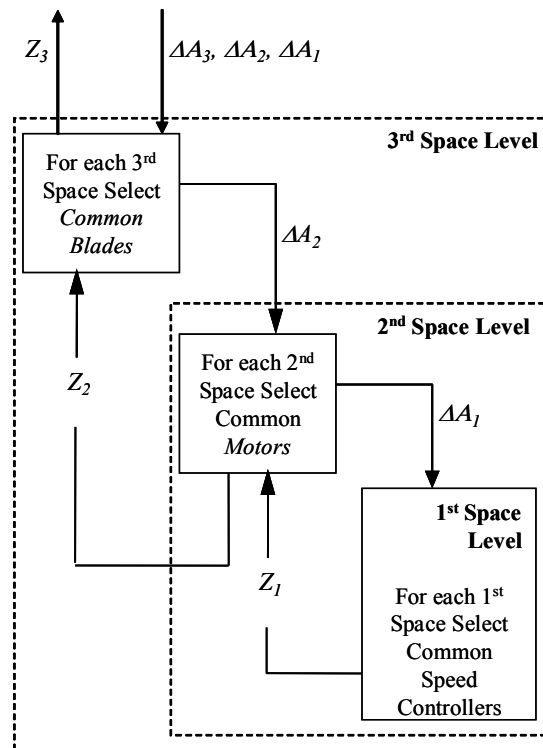
Figure 2.22: The Solution Algorithm for the Fan Example

The objective in this search is the determination of the ranges of each mode for managing variety that provides the lowest average cost of the entire product platform. Results are shown in Table 2.1.

Table 2.1: Results of Fan Example

Decision Variable	Mode of Managing Variety	Range of Commonality
ΔA_1	Customization of air flow capacity using a speed control	48 L/s
ΔA_2	Standardization of motor	96 L/s
ΔA_3	Standardization of blades	192 L/s

Assume for this example that the range of speed control (ΔA_1) found appropriate for changes of airflow capacity is 0 to 48 L/s. In such a case, the market space should then be divided in ten "regions," i.e., ten first space elements: from 20 to 68, 68 to 116 L/s, and so on, as shown in Figure 2.23a.

For the second stage (ΔA_2), assume that two of the smallest space elements is the optimal number to be assembled, i.e., one different motor is to be used for each of the five ranges of airflow of 96 L/s: one motor is used for fans in the range (region or second space element) of 20 to 116 L/s, a second motor for the next range of 116 to 212 L/s, and so on, as shown in Figure 2.23b. Finally, the only mode for managing product variety left is the standardization of blades, and the problem remaining is how many of the five second space elements previously obtained should be "connected" with one blade. For this stage, assume that two is the optimal number. In such a case, there is need for three different blades: two of the blades are used to connect two of the second space elements, and a third one is used for the space left as shown in Figure 2.23c.
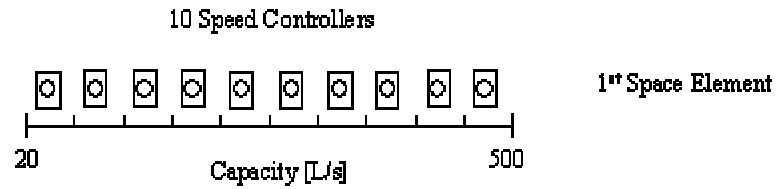
(a) Synthesis of the First Space Element
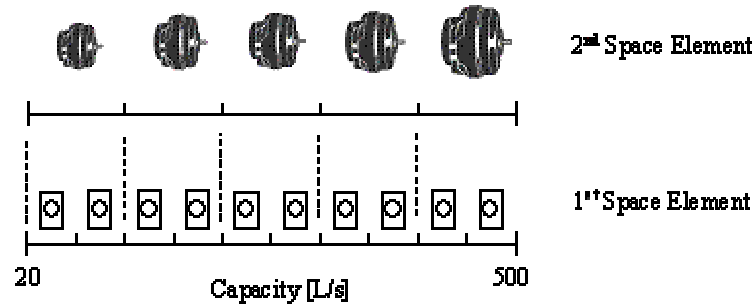
(b) Synthesis of the Second Space Element
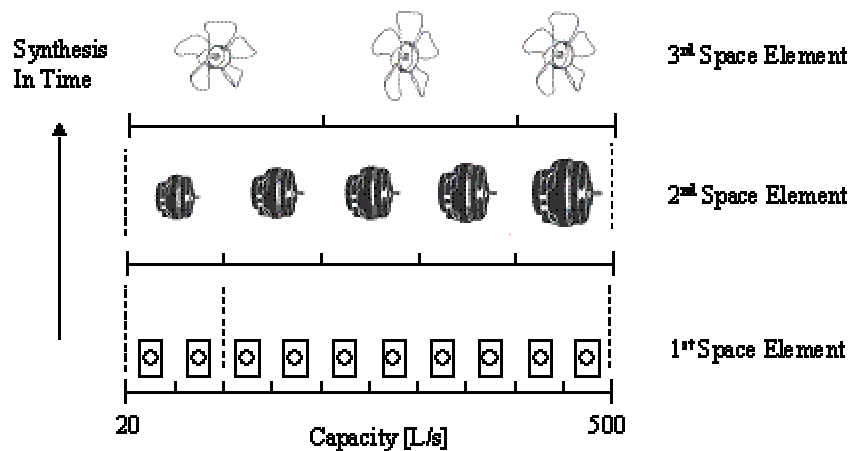
(c) Synthesis of the Third Space Element

Figure 2.23: Hierarchical Arrangement of the Modes of Managing Product Variety for the Fan Example

Using the optimal arrangement and optimal ranges for each mode of managing variety, the specific design parameters for each product variant can be mapped, as shown in Table 2.2.

Table 2.2: Mapping Modes of Managing Variety to Product Specifications

Blades	Motor	Speed Controller	Air Flow
B1	M1	SC 1	20-48L/s
		SC 2	48-96L/s
	M2	SC 3	96-144L/s
		SC 4	144-192L/s
B2	M3	SC 5	192-240L/s
		SC 6	240-288L/s
	M4	SC 7	288-336L/s
		SC 8	336-384L/s

The roadmap of design specification of a section of the entire product family of fans can be seen in the results above. Suppose that a customer orders a fan providing air flow of 130 L/s. From Table 2.2, the manufacturer is informed that by providing speed controller SC 3, motor M1 and number of blades B1 will meet the demand.

Example 2: The resulting three-loop iteration of the variables for the cantilever example is shown in Figure 2.24.
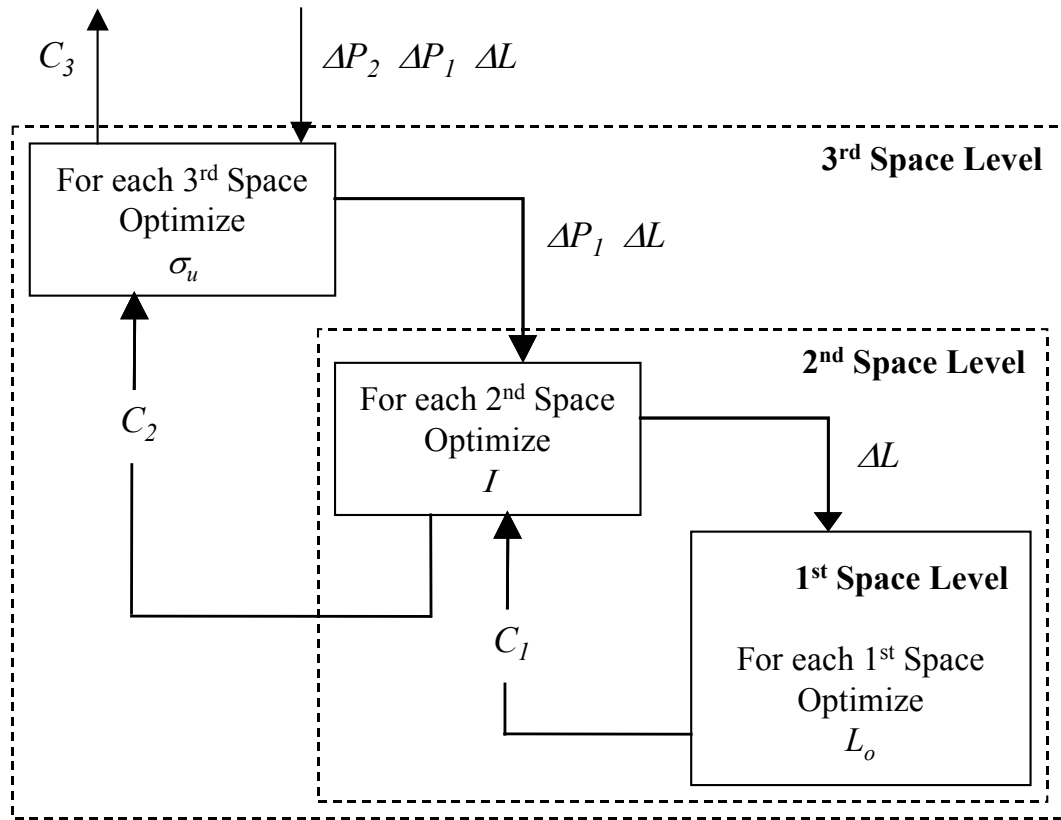
Figure 2.24: The Solution Algorithm for the Cantilever Beam Example (Williams, 2003)

The objective in this search is the determination of the ranges of each mode for managing variety that provides the lowest average cost of the entire product platform. Results are shown in Table 2.3.

Table 2.3: Illustrative Results of Cantilever Beam Tutorial Example

Decision Variable	*Mode of Managing Variety*	*Range of Commonality*
ΔL	Customization of Length of Beam	50 in.
ΔP_1	Standardization of Cross Section	1000 lbs.
ΔP_2	Standardization of Material	2000 lbs.

From the results obtained, designers at BeamTech can now commonalize the raw beam length across a range of 50 in., standardize the cross section across a range of 1000 lbs., and standardize the material type across a range of 2000 lbs.; the cost to manufacture the entire product platform will be at a minimum. The resulting hierarchical arrangement of the three modes of managing product variety is shown in Figures 2.25 and 2.26.
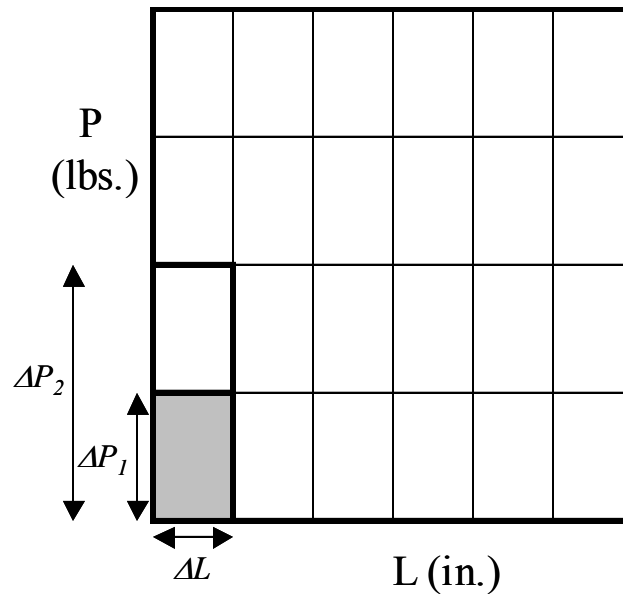
Figure 2.25: Sample Hierarchic Solution of the Cantilever Beam Example
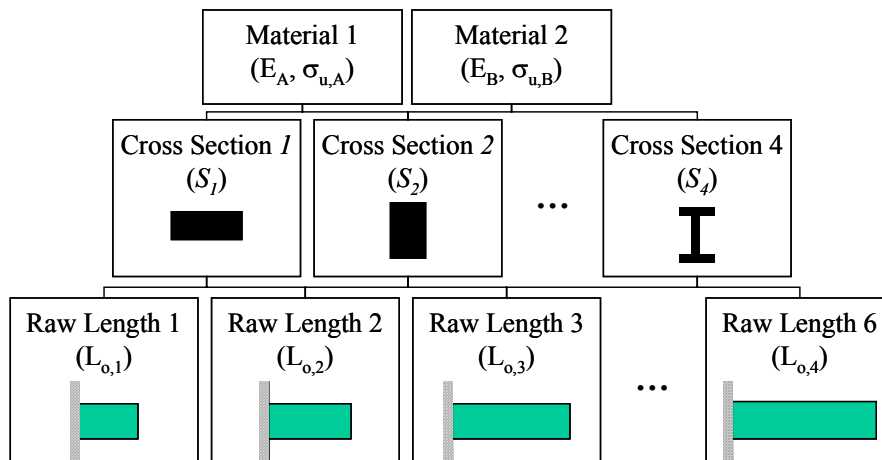
Figure 2.26: Hierarchical Arrangement of the Modes of Managing Product Variety for the Cantilever Beam Tutorial Example (Williams, 2003)

Using the optimal arrangement and optimal ranges for each mode of managing variety, the specific design parameters for each product variant can be mapped, as shown in Table 2.4.

Table 2.4: Mapping Modes of Managing Variety to Product Specifications

L(in.)	P (lbs.)	Lo (in.)	Cross Section	Material Type
150	2000	Lo 1	CS 1	Material 1
	3000		CS 2	
	4000		CS 3	Material 2
	5000		CS 4	
200	2000	Lo 2	CS 1	Material 1
	3000		CS 2	
	4000		CS 3	Material 2
	5000		CS 4	

The roadmap of design specification of a section of the entire product family of fans is seen in the results above. Suppose that a customer orders a beam of length 185 in. that supports a load of 3000 lbs. The results posted in Table 2.4 inform BeamTech that this need is met by cutting a beam of cross-section of CS 2 and material 1 from a raw length of $L_{o\,1}$.

2.1.4   Critical Evaluation of the Product Platform Constructal Theory Method

The Product Platform Constructal Theory Method is based on the foundation of hierarchic systems theory and constructal theory. The concepts of hierarchic systems theory state that a hierarchic method of organization is the most efficient for complex systems. It is possible to synthesize multiple modes of managing product variety for the development of a product platform for customized products; which is typically not

possible in other methods. The resulting hierarchic organization of the modes of managing product variety provides a way to manage the tradeoff between performance and commonality by obtaining the optimal range for each mode.

There are some limitations that have been overcome by the development of PPCTM-PLUS in (Williams et al., 2003). The limitations include inability to deal with multiple objectives, non-uniform demand and no consideration of the manufacturing process. Williams considers non-uniform demand by modeling it as a property of the geometric market space and solving the non-uniform optimal access problem through the discretization of the space. The infusion of the utility-based compromise Decision Support Problem provides a rigorous construct so that decisions with multiple, conflicted, and coupled objectives can be made. The work also applies PPCTM to the development of a "process platform" for the manufacture of customized products. The work in this thesis is largely based on augmenting the PPCTM-PLUS to develop PPCTM-RCM (Robust to Changes in Market).

PPCTM

Preliminary Objectives

☑ Establish fundamental principles for realizing product architectures as open engineering systems, building upon constructal theory.

☑ Develop coherent methodology for systematically designing open product architectures.

☑ Provide a mathematical foundation for supporting decisions made under application of this methodology.

☑ Validate/verify principles

PPCTM-PLUS

Additional Objectives

☑ Non-uniform distribution of demand over market space

☑ Discrete and continuous variables

☑ Process Platforms

PPCTM-RCM

Additional Objectives

☑ Robust to uncertainty in market demand.

☑ Robust to changes in design parameters

☑ Handles changes in extent of market space.

Figure 2.27: PPCTM, PPCTM-PLUS and PPCTM-RCM

However, there are some other limitations that still exist in the PPCTM.

1. *Unable to deal with uncertain demand:* As the market becomes more uncertain, it is difficult to design a product platform. The PPCTM does not cope with such changes in demand. The PPCTM is only able to work if the demand is certain and fixed which is rarely the case.

2. *Unable to deal with changing design parameters:* As time passes, the customer requirements may change as well as there is a need for some flexibility to the

manufacturer. PPCTM does not have any flexibility in the design parameters and the objectives are not met by any change in the design parameters.

3. *No consideration of changing marketplaces:* PPCTM considers the marketplace to be fixed and the product platform is designed for only that market extent. The changing extents of marketplaces are not considered in the PPCTM. However, the manufacturer is bound to explore new areas of the market and capture them. Hence, there is a need for the PPCTM to consider such changes in the extents of the marketplaces.

Example 1: In the case of the fan example, FanTech is able to design a platform for its market using the existing PPCTM but is facing a number of problems after sometime. First, for example, the demand assumed for the market was a snapshot of the probable demand. However, it is observed that the demand changes rapidly and the objectives of the product platform are no longer met due to the uncertain nature of demand.

Secondly, market research observes that the customers are looking for comparatively smaller sizes of fans having nearly the same air flow capacity. However, it is not possible to have such a change in the product platform since it will increase the costs. The reason is that the objectives of the platform are achieved for this particular variety of fans only. So FanTech is looking for some flexibility in the design parameters.

Thirdly, now that the demand in this particular market of air flow capacity has saturated, FanTech is looking for suitable extensions to the existing market. At the same time, it wishes to continue in the existing market. Since it is not economically possible to have a totally new product platform in the adjacent market, the question

---

before FanTech is that how should it extend the existing product platform to the new market and remain profitable.

Example 2: In the case of BeamTech, it is also facing similar problems. It is observed that the demand of the beams fluctuates and providing variety in such circumstances does not meet the objectives of the company. The reason is that the product platform is designed for a fixed demand in the market space. So BeamTech is looking to make its next product platform robust to such changes.

Secondly, the existing product platform specifies the exact cross section that needs to be manufactured. However, if the customer is looking for some other cross section, the manufacturer is not flexible enough to provide him with that particular cross section keeping his other requirements constant. So BeamTech needs some flexibility to such changes in design parameters caused by uncertain factors later on.

Thirdly, BeamTech is right now providing beams with load capacity of 1000 to 5000 lbs. Market research has observed that there is a large demand for beams with higher load capacity. So BeamTech wishes to extend its platform in the adjacent higher load capacity area. By simple extension, BeamTech will not be able to continue to achieve the objectives of the entire new market space. The other option of developing a different product platform in the adjacent area will be very expensive. So the problem that BeamTech is facing is that how it should enter this new market space.

This completes the explanation of PPCTM and its critical evaluation. In the next section, the first theoretical foundation for augmenting PPCTM is discussed.

2.2 ROBUST DESIGN

In the thesis, robust design forms the basis in augmenting the PPCTM. Robust design concepts are used to make the product platform unaffected by variations in demand and design parameters.

Taguchi first proposed the concept of robust design that stressed on improving the quality of a product or process by not only striving to achieve performance targets but also by minimizing performance variation. Taguchi's methods have been widely used in industry (Byrne and Taguchi, 1987; Phadke, 1989) for parameter and tolerance design.

In robust design, a P-diagram (P represents either product or process) as shown in Figure 2.28 is used to show the relationship between different types of design parameters or factors, where (Phadke, 1989).
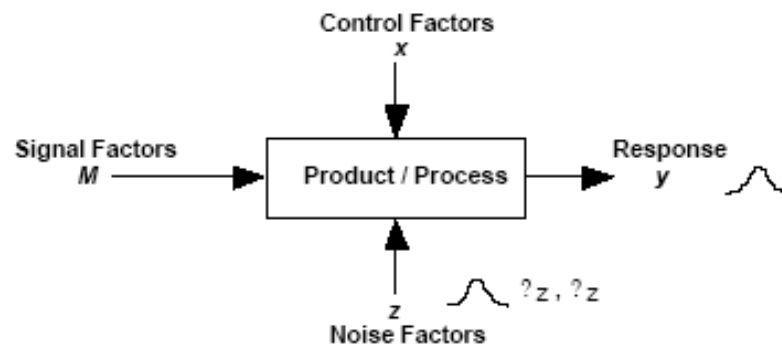
Figure 2.28: P-Diagram of a Product/Process in Robust Design (adapted from Phadke, 1989 – Simpson, 1998)

The three types of factors which serve as inputs to the P-diagram and that influence the (output) response y are as follows:

• **Control Factors** (x) – parameters that can be specified freely by a designer; the settings for the control factors are selected to minimize the effects of noise factors on the response y.

• **Noise Factors** (z) – parameters not under a designer's control or whose settings are difficult or expensive to control. Noise factors cause the response, y, to deviate from their target and lead to quality loss through performance variation.

• **Signal factors** (M) – parameters set by the designer to express the intended value for the response of the product; signal factors are those factors used to adjust the mean of the response but which no effect on the variation of the response.

According to this classification of parameters, in robust design, *x,* is a control factor that can be adjusted by the designer. **M** is the target and *z* is the noise factor. If the only objective is to be as close as possible to *M*, the designs at levels *x*=a and *x*=b are both acceptable because their means are on target. It is observed from Figure 2.29 that when *x*=a the performance varies significantly with the deviation of the noise factor, *z*; however, the performance deviates considerably less when *x*=b. Now considering both objectives of robust design, it is concluded that *x*=b is more robust than *x*=a. By selecting the control factor settings which best minimize the effects of noise on the system performance, it is possible to develop specifications which are robust with respect to a given noise factor (Chen et al., 1996b).
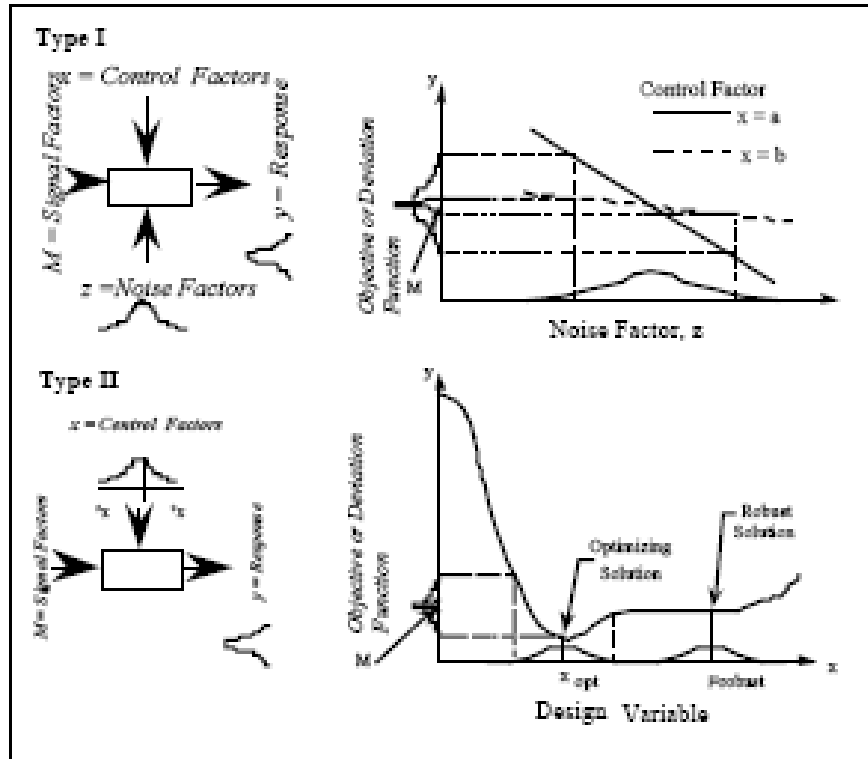
Figure 2.29: A comparison of two types of robust design (Chen et al., 1995)

There are two categories of robust design problems classified in (Chen et al., 1996b). These categories are based on the source of variation:

- Type I – minimizing variations in performance caused by variations in noise factors (uncontrollable parameters)

- Type II – minimizing variations in performance caused by variations in control factor (design variables).

There are a number of limitations to the Taguchi methodology that have been discussed in literature so far. Taguchi has not been able to handle design problems with highly nonlinear behavior accurately (Nair, 1992). It is argued that Taguchi's statistical methods such as orthogonal arrays, linear graphs, etc. are not statistically efficient (Tsui, 1992). Welch et al., (1990) propose to model responses instead of expected loss by

combining the control and noise factors into a single array. Robust Concept Exploration Method (RCEM) (Chen et al., 1996a) has been able to alleviate these limitations. In the RCEM the authors have proposed to integrate the response surface method (RSM) with the compromise Decision Support Problem in developing a general robust design procedure (Chen et al., 1996a).

2.2.1   Robust Concept Exploration Method

The RCEM is a step-by-step approach for the evaluation of design alternatives and generation of top-level design specifications with quality considerations (Chen, et al., 1996a). It facilitates computationally expensive design analysis and design of complex systems. It is the integration of several methods and tools - robust design methods (Phadke, 1989), the Response Surface Methodology (Myers and Montgomery, 1995), Suh's Design Axioms (Suh, 1990) and the compromise DSP (Mistree, et al., 1993).

The method is comprised of four major steps:

1.   *Classify design parameters:* The process starts with the overall design requirements fed as input into Processor A. Each factor is classified into control factors, noise factors and responses. The concept exploration space is then defined.

2.   *Screening Experiments:* Processors B, C and D perform initial screening experiments and analysis of results to fit low order response surface models and reduce the design space. Processor B is the point generator, Processor C is the simulation programs and Processor C is an experiment analyzer.

3.   *Elaborate the Response Surface Model:* This step also involves Processors B, C and D that help perform secondary experiments and analysis of results. The results from the secondary experiments are used to (a) fit second-order response surface models (using Processor E), (b) identify key design drivers and the significance of different design factors and their interactions, and (c) quickly evaluate different design alternatives and answer "what-if" questions in Step 4.

4.   *Generate Top-Level Design Specifications with Quality Considerations:* In this step, top-level design specifications with quality considerations are determined using compromise DSP (Processor F). The response surfaces, functions of control and noise factors, replace the original analyses or simulation programs. Then the compromise DSP is formulated and solved to obtain top-level design specifications.
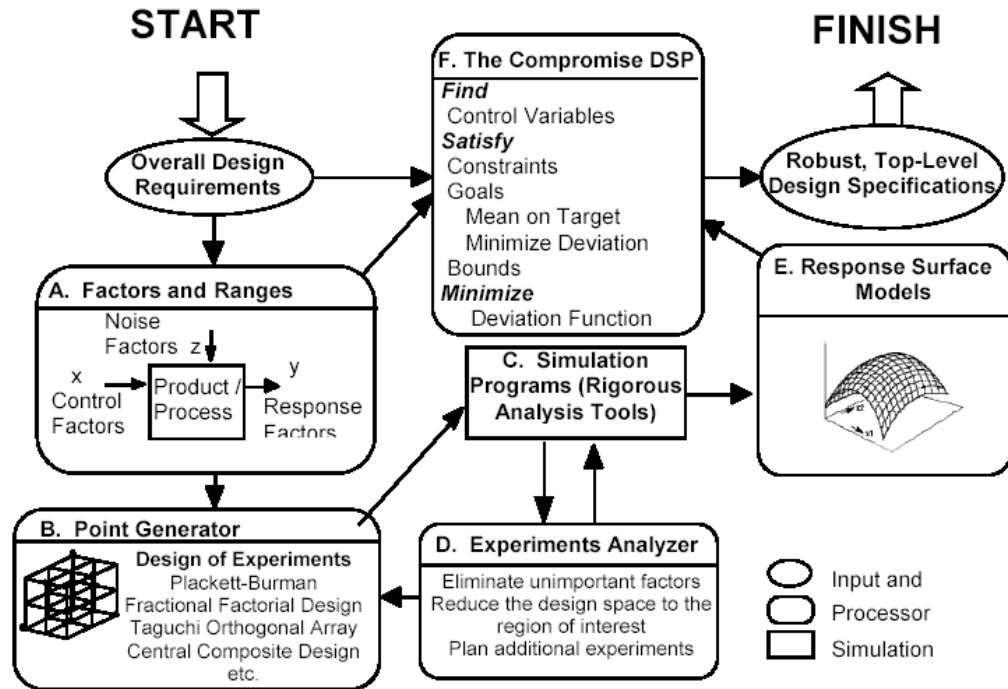
Figure 2.30: Computer Infrastructure of RCEM (Simpson, et al., 1996)

2.2.2 Critical Evaluation of the Robust Concept Exploration Method

There are a number of advantages of the Robust Concept Exploration Method. It makes it possible to have multiple constraints and objectives while also considering the tradeoff between these objectives. Compared to Taguchi's linear model approach, it gives more accurate results. The compromise DSP addresses the issues of bringing the mean to target and minimizing the variation. This makes it possible for the designer to make decisions based on detailed analysis of the system for different values of weights of the objectives.

There are some limitations in the method. The problems of size associated with screening and model building affect RCEM if a complex design problem becomes too large. (Koch, 1997) defines the difference between the class of problems for which RCEM is appropriate, and those which cannot be solved using the method. In the method, the goal and constraint functions are approximated using the methods of statistical design of experiments and specifically quadratic models. Although the approximations have shown to work in a number of engineering problems (Chen, 1995), there are instances in which the performance is highly non-linear and a second-order model is not good enough (Chen, 1996 min.).

Some of the steps in the RCEM have been modified in the thesis to suit the requirements of the problem at hand. In Section 2.3, the next theoretical foundation Compromise DSP has been introduced.

69

2.2.3 Other Approaches in Literature

There are many other approaches in literature that address the dual objective of "bringing mean to target" and "minimize deviation". There are certain drawbacks in the approach of minimizing weighted sums of objectives (Das, I. and Dennis, J.E., 1997). Messac develops the method of physical programming which eliminates the need for weight setting or utility function building in multicriteria optimization (Messac et al., 1996). Chen proposes to use Compromise Programming (CP) (Yu, 1973 and Zeleny, 1973) to address the multiple aspects of robust design (Chen et al., 2000). In some cases in literature, though the multiple aspects of the objective in robust design are acknowledged (Sundaresan et al., 1993), *single* robust design objective function is often utilized. Sundaresan et al. employ a *single* objective function that utilizes *weighting* factors for target performance and variance represented by the Sensitivity Index (SI). Ramakrishnan and Rao, formulate the robust design problem as a nonlinear optimization problem with Taguchi's loss function as the objective (Ramakrishnan and Rao, 1991).

The use of design capability indices for robust design is proposed in (Chen et al., 1999). This is one of the approaches analyzed by the author in this section to design robust product platforms. Similar difficulties as stated in the discussion later, on using design capability indices to design a robust product platform, may have to be faced in most other approaches for robust design.
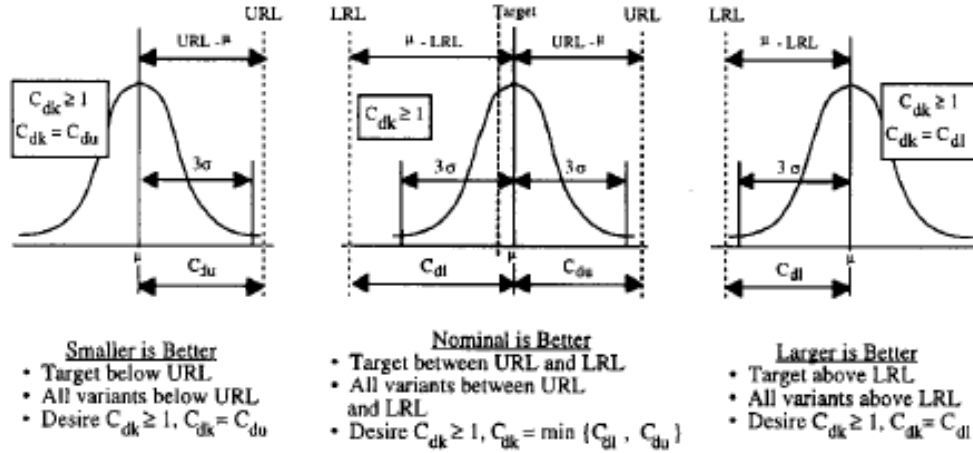
$$C_{dl} = \frac{\mu - LRL}{3\hat{\sigma}}; \quad C_{du} = \frac{URL - \mu}{3\hat{\sigma}}; \quad C_{dk} = \min\{C_{dl}, C_{du}\}.$$

Figure 2.31: Design Capability Indices (Chen et al., 1999)

Design capability indices are based on process capability indices from Statistical Process Control (SPC), an approach for quality control rooted in manufacturing. A stable process is represented by a measure of its variation; six σ standard deviation is commonly chosen. The simplest measure of process capability, Cp, compares the variation of a process to the customer specifications through the equation:

$$C_p = \frac{USL - LSL}{6\hat{\sigma}}$$

Where USL is the upper specification limit, LSL is the lower specification limit, and σ is the estimated standard deviation of the process. Although C_p is a measurement of the spread of the process in relation to specification width, it does not measure how well the process average is centered about the target value. The process capability indices C_{pl}, C_{pu} and C_{pk} are used to measure both process variation with respect to customer specifications and the location of the process average. C_{pl} and C_{pu} are process capability

71

indices for single-sided specification limits while C_{pk} measures process capability for two-sided specification limits. The process capability indices, C_{pl}, C_{pu} and C_{pk}, become design capability indices C_{dl}, C_{du} and C_{dk} which measure the portion of the range of designs that satisfies the ranged design requirement (Chen et al., 1999). In Figure 2.31, the upper and lower specification limits used with C_{pk} are replaced by upper and lower requirement limits (URL and LRL, respectively) since the emphasis is on satisfying design requirements and not machining specifications. Consequently, C_d, which is based on C_p and does not account for the mean being off target, is not used as a design capability index; only C_{dl}, C_{du} and C_{dk} are used for different design situations.

Now consider applying the concepts of design capability indices in the product platform design problem. Assuming the goal of maximizing an objective of the product platform, the rightmost part of the figure is considered where 'larger is better'. Here, designs with a $C_{dk} \geq 1$ are capable of meeting this requirement, and C_{dk} is equal to C_{dl} as shown in Figure 2.31. Distributions with a $C_{dl} < 1$ will have designs which do not meet the robust design requirement.

However, to apply design capability indices to the PPCTM, there are three main difficulties:

1. In a product platform problem, there are a number of products in the marketplace. For every product that is present, there is a need to check robustness. In the case of design capability indices this means that for every product the corresponding design capability index is determined. So one has to compare every product platform comprising of products, each having an index, with other possible product platforms. In the current approach, the principle of summation of

variances helps to get a quantity that considers the variation of the entire product platform. However, using design capability indices it is not possible to get a quantity for every product platform that can be compared; individual indices for every product are only obtained.

2. In a product platform design using PPCTM, at the beginning, one does not know how many products are present in the product platform. So predicting the upper and lower requirement limits for every product is difficult. This is needed to find out the index for every product.

3. The indices can be used to a certain extent where one can check for all the products in the product platform if $C_{dk} \geq 1$. If yes, the product platform is achieving the robust design objectives. However, it is not possible to find the most robust product platform if there are more than one product platform satisfying $C_{dk} \geq 1$.

So based on these difficulties one can say that using design capability indices in PPCTM may not be feasible.

For modeling the designer's preference structure regarding the two aims of robust design, different methods have been proposed, which follow different paradigms for multiobjective decision making. Iyer and Krishnamurty (1998) present a preference-based metric for robust design using concepts from utility theory (von Neumann and Morgenstern, 1947; Keeney and Raifa, 1976; Hazelrigg, 1996) to capture the designer's intent and preference when making the tradeoffs between mean and variation of performance. Utility theory can be another approach to consider uncertainty. Under the notion of utility theory, the ultimate overall worth of a design is represented by a

multiattribute utility function which is assumed to incorporate designer intent. The major difficulty while using a utility function is it is often impossible to obtain a reliable mathematical representation of the decision-maker's actual utility function.

Steuer writes regarding utility functions (Steuer, 1989 – p. 4) "One expects [the utility function] to be nonlinear. However, this is not the main difficulty with this approach. The main difficulty is that with many problems it is not possible to obtain a mathematical representation of the decision maker's utility function U. It is about such problems that we are concerned." Steuer further writes (Steuer, 1989 – p. 146) "In this book, we are assuming that, in practice, we will never know a mathematical representation of the DM's utility function U." These comments point to the lack of practical usefulness of utility theory in the effort of uncovering the appropriate utility function (Chen et al., 2000).

Based on this, it is the author's opinion that control of the weights of the two objectives of robust design is more effective than relying on the mathematical representation of the utility function in a product platform design problem. However, it would be interesting to compare the results obtained with results from other approaches.

2.3 THE COMPROMISE DECISION SUPPORT PROBLEM (C-DSP)

The compromise DSP is a multiobjective decision model that is a hybrid formulation based on Mathematical Programming and Goal Programming (Mistree, Hughes et al., 1993). It is used to determine the values of design variables that satisfy a set of constraints and achieve a set of conflicting goals as well as possible. In general, the compromise DSP facilitates the improvement of a feasible alternative through

modification. The compromise DSP is similar to goal programming. In a compromise DSP, the objective is to minimize the deviation function, which is a function of the goal deviation variables. System goals are formulated in terms of system and deviation variables. On the other hand, in traditional mathematical programming objectives are functions of the system variables only. However, the compromise DSP retains the system constraints employed in traditional optimization.

The mathematical form of the compromise DSP is shown in Figure 2.32. The system descriptors, namely, system and deviation variables, system constraints, system goals, bounds and the deviation function are described in detail in (Mistree, Hughes et al., 1993). *System variables* $(X_1, X_2, \ldots, X_n)$ are independent, in control of the designer and usually describe attributes of the system. A Compromise DSP formulation must have at least two system variables that may be continuous, discrete, and/or Boolean variables. *Bounds* help to represent the region in which a search is to be made for a feasible solution. Each of the system and deviation variables has a lower an upper bound associated with it. *System constraints* model the relationships between capabilities of the system and demands placed upon the system. Feasible designs always satisfy the set of system constraints; thus defining the feasible design space. System constraints are functions of the system variables only. *System goals* are used to model the aspiration a designer has for a system. They relate the goal (aspiration level), G_i, of the designer to the actual attainment $A_i(X)$ of the goal. The *deviation variables* d_i^-, d_i^+ indicate the extent to which the goals are achieved. d_i^- and d_i^+ represent the level of underachievement and overachievement of a goal, respectively. Two constraints as shown in Equation are

placed on deviation variables to ensure that the deviation variables are never negative and

that one of the deviation variables will always be zero.

$$d_i^- \bullet d_i^+ = 0 \text{ and } d_i^-, d_i^+ \geq 0 \qquad\qquad [2.1]$$

Given

    An alternative to be improved through modification.
    Assumptions used to model the domain of interest.
    The system parameters.
    All other relevant information.

        n   number of system variables
        p+q   number of system constraints
        p   equality constraints
        q   inequality constraints
        m   number of system goals
        $g_i(\underline{X})$ system constraint function
        $f_k(d_i)$ function of deviation variables to be minimized at Priority level k for the preemptive case
        W_i weight for the Archimedean case

Find

    The values of the independent *system variables*
        $\underline{X} = X_1, \ldots, X_j$ j = 1, …, n
    The values of the *deviation variables*
        d_i^-, d_i^+ i = 1,…, m

Satisfy

    The *system constraints* that must be satisfied for the solution to be feasible. There is no restriction placed on linearity or convexity.
        $g_r(\underline{X}) = 0$ r = 1,…,p
        $g_r(\underline{X}) \geq 0$ r = p+1,…, p+q
    The *system goals* that must achieve a specified target value to the extent possible. There is no restriction placed on linearity or convexity.
        $A_i(\underline{X}) + d_i^- - d_i^+ = G_i$;   i = 1,…,m   [Eqn 2.2]
    The lower and upper bounds on the system.
        $X_j^{min} \leq X_j \leq X_j^{max}$;   j = 1,…,n
        $d_i^-, d_i^+ \geq 0$ and $d_i^- \cdot d_i^+ = 0$

Minimize

    The *deviation function* (a measure of the deviation of the system performance from that implied by the set of goals and their associated priority levels or relative weights):
        Case a: Preemptive (lexicographic minimum)
        $Z = [f_1(d_i^-, d_i^+), \ldots, f_k(d_i^-, d_i^+)]$ [Eqn 2.5]

        Case b: Archimedean
        $Z = \sum_{i=1}^{m} W_i(d_i^- + d_i^+)$; $\sum W_i = 1; W_i \geq 0$; i= 1, …, m   [Eqns 2.3,2.4]

Figure 2.32: Mathematical Formulation of the Compromise DSP (Mistree, Hughes, et al., 1993)

The fundamental idea behind the compromise DSP is to minimize the difference between that which is desired and that which can be achieved. Hence the formulation minimizes the deviation function, $Z(d_i^-, d_i^+)$, which is in terms of deviation variables. Smaller the value of the deviation function, closer is the goal to the ideal value. Since multiple deviation variables are combined in the deviation function, it is desirable to have the deviation variables of the same order of magnitude. This is achieved by normalizing the goals before combining them into a deviation function. A set of rules has been established for formulating the system goals in a manner such that the deviation variables will be of the same order of magnitude.

1) If a designer wishes to maximize achievement, $A_i(\underline{X})$, he/she chooses a target

    value G_i greater than or equal to the maximum expected value of $A_i(\underline{X})$.

    Thus, the ratio $A_i(\underline{X})/G_i$ will always be less than or equal to 1, and the

    system goal may be formulated as follows:

$$A_i(\underline{X})/G_i + d_i^- - d_i^+ = 1 \qquad\qquad [2.2]$$

    where d_i^+ will always be zero and the objective is to minimize the

    underachievement d_i^- to ensure that the design is as close as possible to

    the goal.

2) If a designer wishes to minimize achievement of a goal, he/she chooses a

    target value G_i less than or equal to the minimum expected value of $A_i(\underline{X})$.

    The system goal becomes: $G_i / A_i(\underline{X}) + d_i^+ - d_i^- = 1$ where d_i^- will always be

    zero, and the objective is to minimize overachievement, d_i^+ to ensure that

    the design is as close as possible to the desired goal.

3) If a designer wishes to achieve a target value, he/she may use formulation 1 if the target value is approached from below by $A_i(\underline{X})$ or formulation 2 if the target value is approached from above. In both cases the sum $(d_i^- + d_i^+)$ is minimized.

The use of compromise DSP in making a tradeoff between multiple conflicting goals makes it effective in having a compromise between different objectives in product family design. Similarly, compromise DSP has been used in several problems of robust design. The conflicting goals in robust design are "bringing the mean to target" and "minimizing the variance"; compromise DSP is used to make a tradeoff between these goals. So it manages to have a tradeoff between obtaining the objectives as well as minimizing the deviation of the objectives in the presence of noise.

Critical Evaluation of the Compromise DSP

As discussed in Section 2.3, the compromise DSP is a flexible decision support construct that facilitates the search for satisficing compromises among multiple, conflicting goals. It also accommodates multiple constraints and bounds on the system variables and implementable with reasonable effort. It is domain independent. Also, the compromise DSP is applicable along a design timeline, including during the early stages of design or under other conditions when decisions must be made quickly and/or with limited information. The compromise DSP has also been used for the design of product platforms (Seepersed et al., 2002). In addition, Archimedean deviation functions are useful for exploring tradeoffs among multiple goals, and preemptive deviation functions do not require weights—an advantage in the early stages of design.

The limitations of the baseline compromise DSP include iterative weight setting and sensitivity to goal target values. In addition, for considering uncertainty with respect to goal values, modifications need to be made to the baseline compromise DSP. Using weights or priority levels, there is no rigorous, axiomatic way to quantify and consider designer preferences for uncertain goal values (Seepersed, 2002).

2.4 STRATEGIC DESIGN

The PPCTM has been augmented in Chapter 5 considering the objectives of strategic design.

The concept of 'strategic design' was first introduced in (Seepersad, Cowan, et al., 2002). The subtle difference between design and strategic design was explained in Chamberlain, 2002. Strategy involves positioning resources ahead of time, or "prior to actual engagement." Most design is concerned with fulfilling specific customer requirements.

In some design practices, changes in those requirements are taken into account by making products flexible, customizable, or modular. Few if any methods take specific changes in requirements into account. Strategic design is meant to take those specific changes over time into account, forecasting trends and taking action in accordance with them well in advance. Chamberlain in (Chamberlain, 2002) recognizes that the company should develop forecasts of future market trends to make sure that the money invested in a new product platform or in the development of a new piece of technology continues to pay dividend when the next big market shift occurs a few months or years down the line. All of the actions that the company needs to take at the current time to recognize and

account for future trends fall under the general heading of an activity that is defined as strategic design.

Seepersad and coauthors have proposed the following definition of strategic design (Seepersad, Cowan, et al., 2002):

"Strategic design is a comprehensive approach for forecasting shifts or changes in markets, associated customer requirements, and technical capabilities, and for devising artifacts that accommodate these shifts efficiently and effectively. It is a marriage of strategic product planning and market analysis, methods for leveraging and adapting existing products, procedures for assessing and infusing technological innovations, and systematic evaluation techniques for comparing and selecting among a portfolio of options."

The strategic design process is divided into four steps. Each of these steps involves a number of activities to be carried out (Seepersad, Cowan, et al., 2002):

Step 1 – Identification of Current Product Capabilities

This involves identifying relevant capabilities and characteristics of the existing products. This introspection helps to know at what stage is the design capability currently.

Step 2 – Identification of Need and Opportunities

This step involves the design team to look at opportunities for the future. This is done by careful analysis of market trends and customer demands and assessing what opportunities exist for expanding the market coverage of its current product line. Current

trends and future trends from extrapolation need to be studied. This would in turn involve prioritizing the projects to discard the not so favorable projects and picking up the favorable ones.

Step 3 – Identification of Strategies for Fulfillment of Market Demand

Now, with the capabilities and need identified, it is important to identify the different options available for the design team. The options could include simple redesigning of existing products using current technology, redesigning the products with existing technology or investing in disruptive technologies that may get good returns by expanded product capabilities in the long run. Hence, in this step the team must identify such strategies for fulfillment of market demand.

Step 4 – Selection of Technology Options

After identifying a number of feasible strategies, the next step is to select the best using criteria that takes into account costs, performance, time, and any changes in performance over the useful life of the product. This involves using a systematic evaluation and selection method. The team must also have the steps for implementing the selected option planned out.

The augmented steps in the PPCTM involve exploring potential portfolio expansions and considering its objectives, which help to take action in accordance with the future well in advance. Relating to strategic design concepts, this makes sure that the money

invested today is made worth of by getting dividends even when the next big market shift occurs in future.

2.5 A LOOK BACK AND A LOOK AHEAD

In Chapter 2, the author broadly presents two important discussions in the thesis: the presentation, discussion and critical analysis of (1) the Product Platform Constructal Theory Method, and (2) the constructs that will be used to augment this method to enable it to cope with uncertain demand, changing design parameters and changing extents of marketplaces. Hence it establishes partly the Theoretical structural validity for Hypotheses 1, 2, and 3 through the discussion of the advantages, limitations, and accepted domains of the constructs.

Constructs	Building Blocks	References of Building Blocks	References of Constructs	Descriptions	Critical Evaluation
		Are the constructs themselves valid? Have they been cited in literature before?			
Product Platform Constructal Theory Method	Hierarchic Systems Theory, Constructal Theory	Simon, et al., 1961, 1996 Bejan, 1996, 1997, 2000	Hernandez, 2000, 2001, 2002, 2003 Williams, 2003 Carone, 2003	Section 2.1.1, 2.1.2, 2.1.3	Section 2.1.4
Robust Concept Exploration Method	Robust Design Methods Response Surface Methodology Suh's Design Axioms Compromise DSP	Phadke, 1989 Myers and Montgomery, 1995 Suh, 1990 Mistree, et al., 1993	Chen, et al., 1995, 1996a,b Mistree, et al., 2002 Simpson, et al., 1996, 1998	Section 2.2.1	Section 2.2.2
Compromise DSP	Mathematical Programming Goal Programming	Charnes and Cooper, 1977	Mistree, Hughes et al., 1993 Seepersad, 2000 Fernández, 2001	Section 2.3	Section 2.3.1
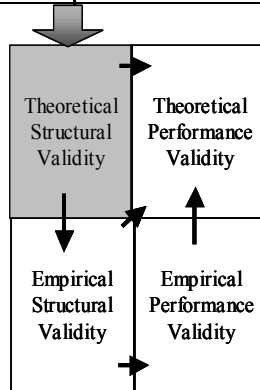
Figure 2.33: Theoretical Structural Validity

The Product Platform Constructal Theory Method is presented in Section 2.1. This involves first discussing the hierarchic systems theory and constructal theory. Based on this discussion, Hernandez's abstraction into the realm of product family design is presented in Section 2.1.2. In Section 2.1.3, the six steps of the PPCTM are presented. The discussion of the PPCTM is ended by its critical analysis in Section 2.1.4. The limitations existing in the PPCTM such as the inability to consider uncertain demand, changing design parameters and changing extents of marketplace are presented in this section. The alleviation of these limitations forms the basis of the thesis.

The Robust Concept Exploration Method is presented in Section 2.2, which will address the limitation of uncertain demand and changing design parameters. The compromise DSP, discussed in the next section also plays an important role in alleviation of the aforementioned limitations as well as the problem of changing extents of marketplaces. The idea of strategic design which forms the basis of considering changing extents of marketplaces while designing is discussed in Section 2.4. Each of the constructs is discussed in detail along with its critical analysis at the end.

In Chapter 1, the need and motivation for developing a systematic approach for the design of product platforms that alleviates the existing limitations is presented. In this chapter, what are the constructs that will help to alleviate the limitations are discussed. The method developed using the constructs discussed in this chapter is presented in Chapter 3. The application of the augmented method is applied to two example problems in Chapters 4 and 5.

CHAPTER III

A METHOD FOR DESIGNING PRODUCT PLATFORMS FOR UNCERTAIN DISTRIBUTION OF DEMANDS, CHANGING DESIGN PARAMETERS AND EXTENTS OF MARKETPLACES

The author's primary objective in this thesis is the augmentation of the Product Platform Constructal Theory Method. This augmentation is achieved by extending the formulation and the approach of the PPCTM to overcome some of its limitations. The aim is to answer the primary research question posed in Section 1.3.

> *How should the Product Platform Constructal Theory Method be augmented to deal with the uncertainty in distributions of demand, changing design parameters and extents of marketplaces?*

As is seen from the research question, the author wishes to augment the PPCTM so that it handles uncertain distribution of demand, changing design parameters and extents of marketplaces. In this competitive environment of customization, there is a need to deal with uncertain distribution of demand and changing extents of marketplaces. At the same time, there are changes in customer requirements that make changing design parameters during the product life cycle necessary. These three major research thrusts (uncertain distribution of demand, changing design parameters, and changing extents of marketplaces) are the focus in the augmentation of the PPCTM.

In Section 3.1 of this chapter, integration of aspects of RCEM into PPCTM that make it possible to design a product platform robust to changes has been presented. In the Section 3.2, the steps that should be combined into the PPCTM to consider changing extents of marketplaces have been presented. In the last Section 3.3, the steps of previous two sections have been integrated into the PPCTM and the augmented PPCTM is presented.

At the end of this chapter, theoretical structural validity is provided to show internal consistency of infusing the proposed augmentations into the PPCTM. The author's role in this chapter is to provide theoretical structural validity as shown in the Figure 3.1 and Figure 3.2.
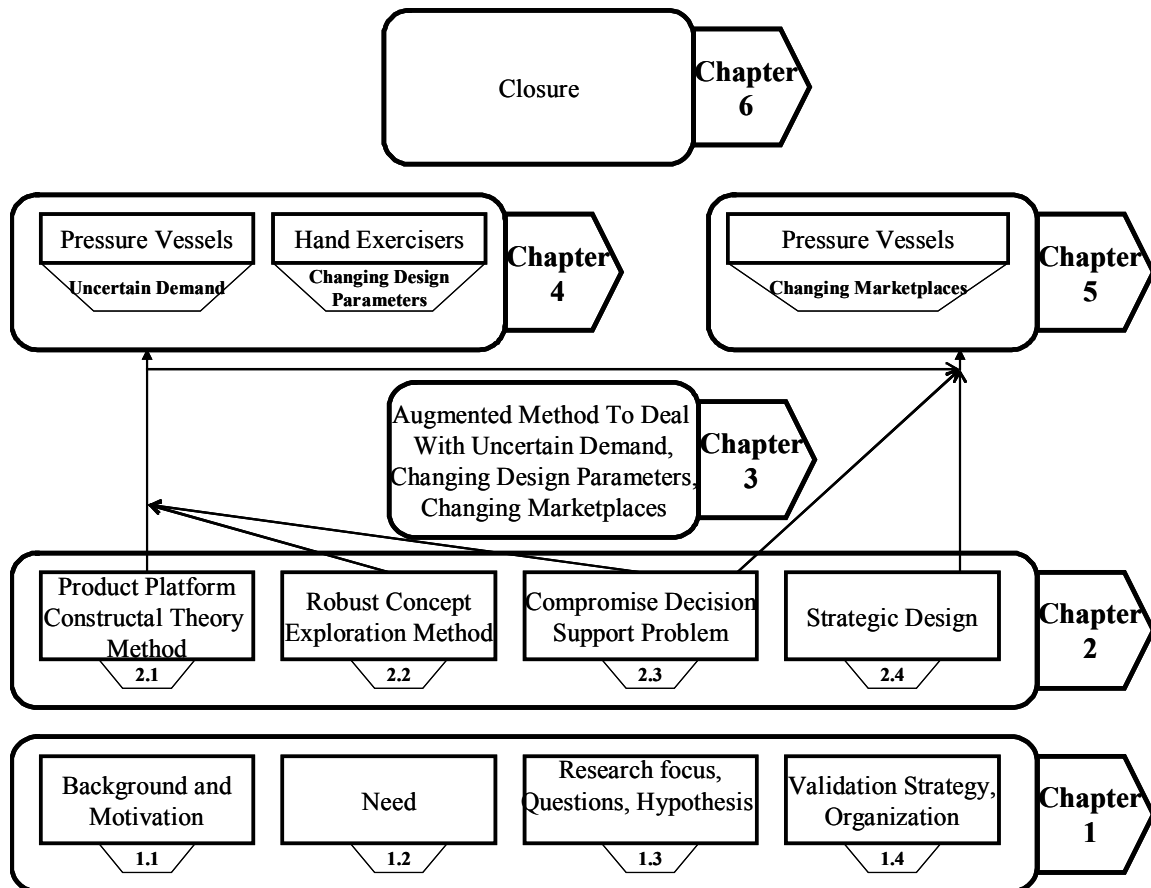
Figure 3.1: Overview of chapters

		Relevance	**Hypotheses**

Chapter 1
Designing Product Families for Mass Customization

- Introduction, Motivation and Frame of Reference
- Research Questions and Hypotheses
- Organization of the thesis

Introduce

Chapter 2
The Theoretical Foundation

- Theoretical foundation of PPCTM
- Detailed description of Hernandez's PPCTM
- Description of constructs of potential augmentations
- Theoretical structural validation

Elaborate

Chapter 3
A Method for Designing Product Platforms to Deal with Uncertain Demands, Changing Design Parameters and Changing Marketplaces

- Infuse concepts of robustness into product platform design
- Use compromise DSP for strategic product platform design
- Description of the method
- Theoretical structural validation

Method for Implementation of Hypotheses

Chapter 4
Customized Product Platform Design in an Uncertain Environment: Pressure Vessels and Hand Exercisers

- Demonstrate that the proposed augmentation is able to deal with changes in demand and design parameters.
- Empirical Structural and Performance Validity

Verify H1

Chapter 5
Customized Product Platform Design for a Changing Marketplace: Pressure Vessels

- Demonstrate that the use of compromise DSP is an effective way of strategically designing product platforms for a changing marketplace.
- Empirical Structural and Performance Validity

Verify H2

Chapter 6
A Look Back and A Look Ahead

- Summarize and critically review research findings
- Contributions and limitations
- Recommend future work
- Theoretical performance validation

Summarize

Foundation / Problem Identification

Method

Hypothesis Testing

Closure

Figure 3.2: Thesis Roadmap
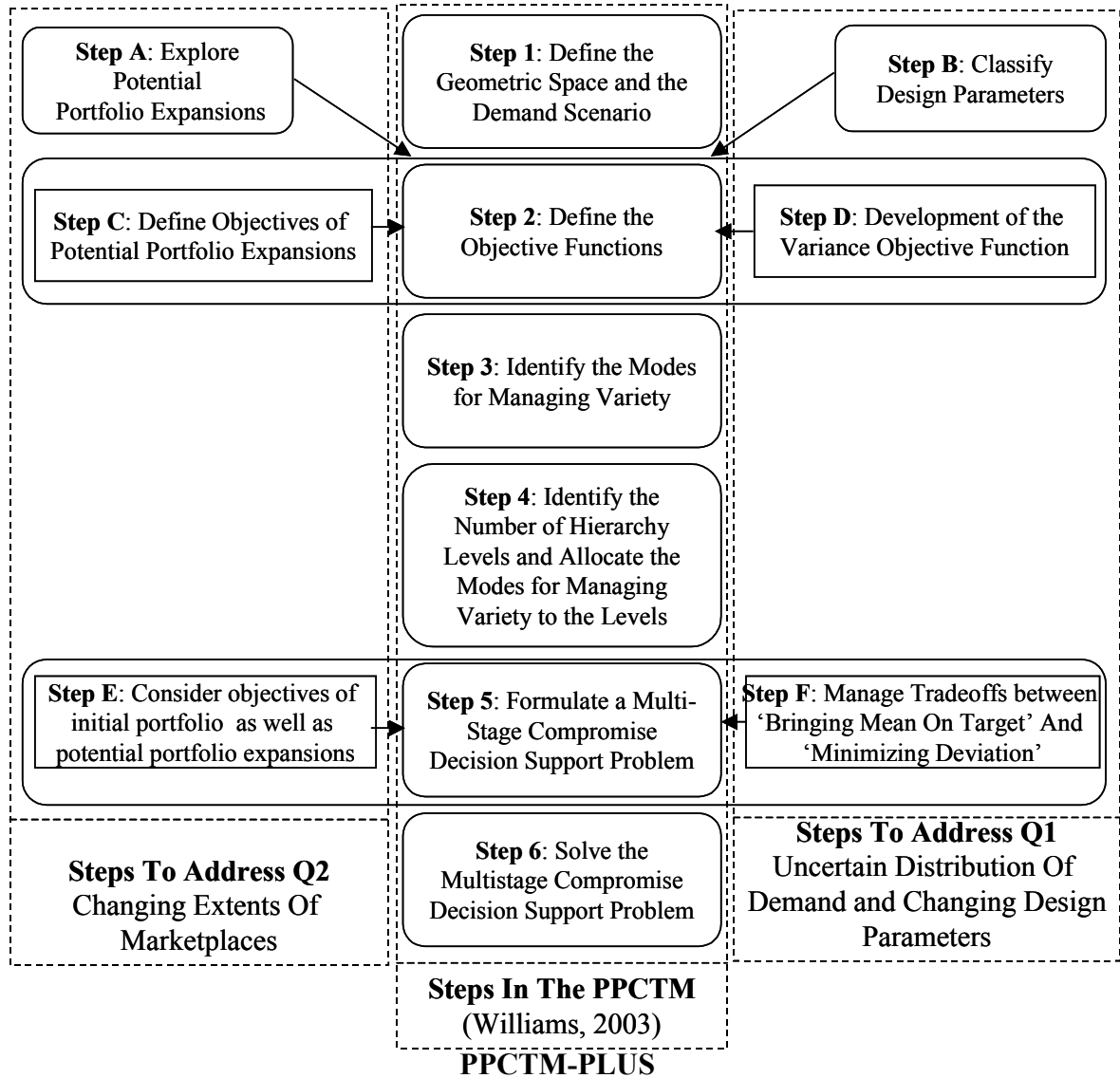
Figure 3.3: Augmentations to the PPCTM

3.1 THE INFUSION OF ROBUST DESIGN PRINCIPLES TO HANDLE UNCERTAIN DISTRIBUTION OF DEMAND AND CHANGING DESIGN PARAMETERS

In this section, the author focuses on the steps that make the PPCTM robust to uncertain distributions of demand and changing design parameters. These steps can be added or removed from any PPCTM module as and when robustness needs to be

considered. The proposed approach is to integrate RCEM and the PPCTM to make the entire process robust to changes. The explanation of the integrated PPCTM and RCEM is presented below.
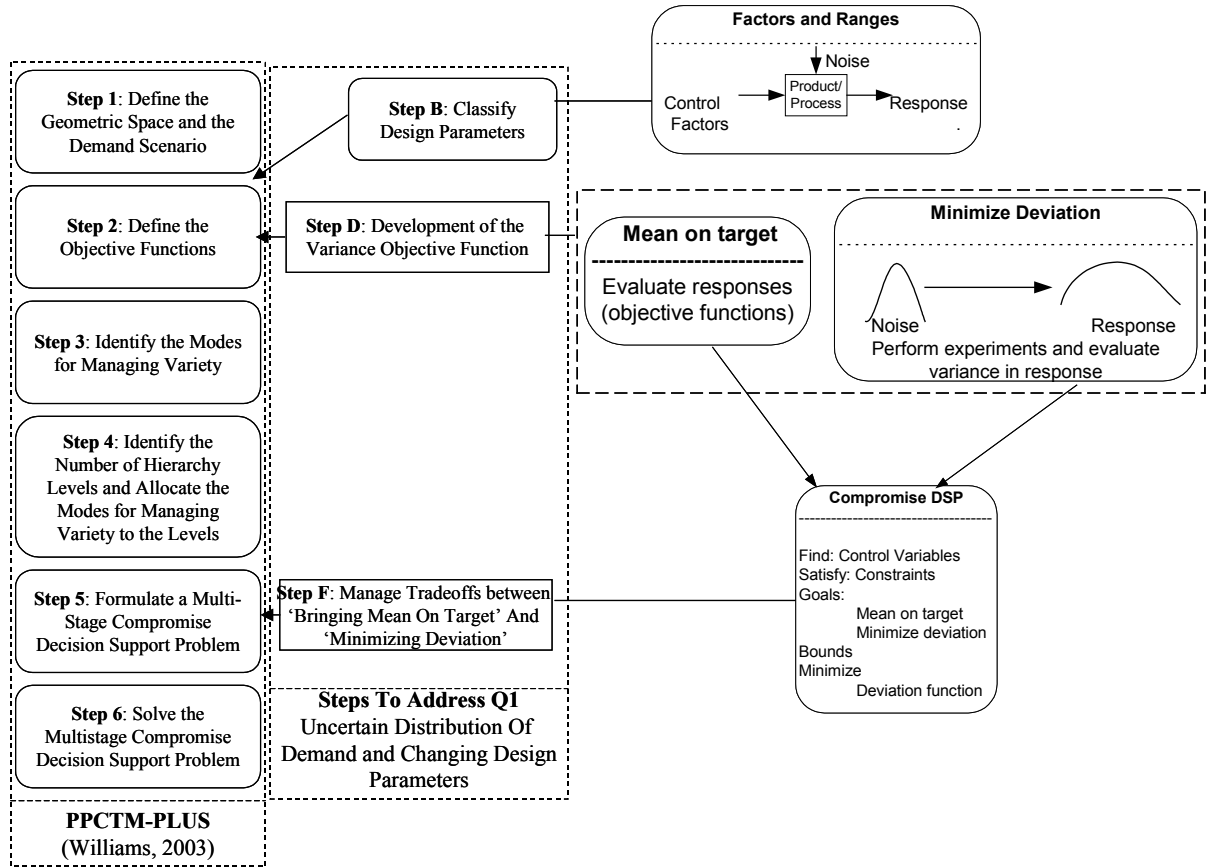
Figure 3.4: Infusion of Robust Design Principles

The infusion of robust design principles into the PPCTM is shown in Figure 3.4. The market space, demand scenario and objectives are given as input to PPCTM. In Step B, the factors are classified and used in the next step. PPCTM with the goal to achieve mean on target has been developed in PPCTM-PLUS (Williams, 2003). The other goal of minimizing the variation in objectives using PPCTM is explained in Step F that gets integrated into the step 'Formulate a Multistage Compromise Decision Support Problem'.

Compromise Decision Support Problem is used to resolve the tradeoff between the two conflicting goals. Each of these augmented steps is explained in detail in the next part.

Steps to be Incorporated for Robust Design of Product Platforms (See Figure 3.4)

<u>**Step B - Classify Design Parameters (Module A – Section 2.2)**</u>

As is seen in Figure 3.4, after defining the geometric space and demand scenario, this step is needed. This step remains the same as in the Step 1 (module A) of RCEM. In this step, the initial concept exploration space is defined and the problem is formulated as a robust design. Design variables are grouped as either control factors or noise factors. Ranges of values are specified. The responses, i.e., objective functions that need to be studied are also identified.
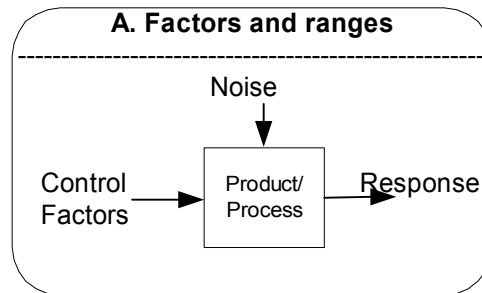
Figure 3.5: Parameter diagram

The method facilitates handling of both the types of robust design problems: Type 1 and Type 2 (discussed in Section 2.2). In the method, Type 1 problem involves demand as a noise factor, the product specifications as the control factors and the response as the objectives of that platform. The Type 2 problem involves deviation of the design parameters that are the control factors.

Thus, in this step the classification of terms in the product platform is done.

As discussed in Section 2.1.4, BeamTech is facing the problem of uncertain distribution of demand. So it has decided to rectify the problem for its next product platform offering. To do this, in this step, it needs to first classify its design parameters. The classification is shown in Figure 3.6.
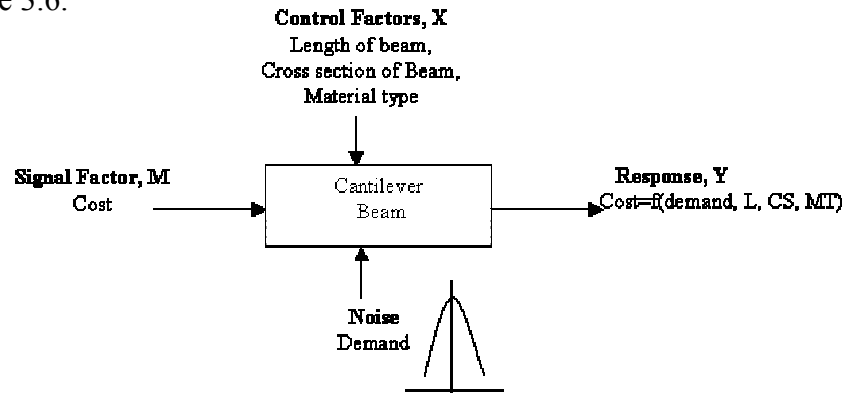
Figure 3.6: Classification of Design Parameters of Cantilever Beam Example

This is a Type 1 robust design problem. Demand is considered as the noise.

PPCTM Step 2 - Define the objective functions (Development of the variance objective function) (See Figure 3.4)

The existing step of PPCTM also needs to include the development of the variance objective function. This is explained in detail below:

Perform experiments and evaluate variance in response (Module C – Section 2.2):

This part is fundamentally the same as Step 2 and 3 (module B, C, D, E) of RCEM. However, it is modified significantly to suit the requirements.

The steps of RCEM need to be modified due to the problem of size (discussed in Section 2.2.2). Efficiency of exploration with RCEM is made possible by incorporating approximate models. The experimentation necessary for building these models requires fewer number of design variables or else efficiency and accuracy are sacrificed. Consider an experiment for building a second order response surface model in 10 factors. A full *central composite design* (CCD) for 10 factors requires 1045 experiment cases (Koch,

1997). In PPCTM, the entire market space is comprised of a number of products. Every product has a particular set of control factors. Since the number of products in the family is large, the number of levels required for each factor is large. So the number of experiments to be performed is very high. Moreover, the response surface model developed from the results of these experiments performed is not accurate enough to give any useful information. This in turn hinders further calculations of finding the variance of the model. Hence, alternate steps need to be looked into to suit our requirement.

Now, the function of these steps in RCEM is noted. Typically, these steps of RCEM help in developing system goals for "moving the mean to target" and "minimizing variation". As discussed previously, the goal of "moving the mean to target" has already been done in (Williams, 2003). It is suggested that the development of the other goal of "minimizing variation" could also be achieved by using the PPCTM. The development of this goal of "minimizing variation" is explained below:

Development of the variance objective function:

For every product that is considered, some amount of noise (in case of demand) is provided. The amount of noise that is provided can have any value. The variation in response due to the noise is noted. From these values of responses, the variance in response is calculated.
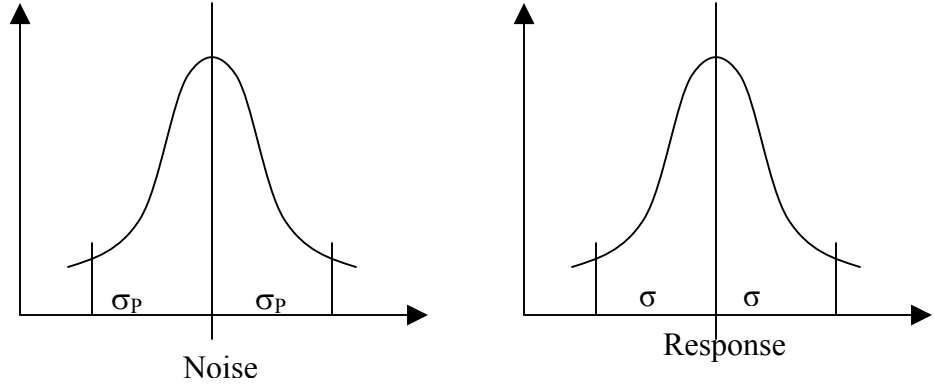
Figure 3.7: Noise and Response – Mean and Standard Deviation

So, for a product (keeping all other design variables constant) with certain values of noise, the corresponding variance in response is calculated. Hence for every product considered in the PPCTM, corresponding variance in response is calculated. The objective is to minimize such variance. The variance in response is calculated by the following formula (Hayter, 2002):

$$\sigma^2 = \sum (R_i - R)^2 / (n\text{-}1) \qquad\qquad [3.1]$$

Where R_i is the response calculated after giving variance σ_P to parameter P_i

R is the response at mean level of parameter P_i

n is the number of levels of the noise factor

Considering the entire market space, the sum of variances of all considered products in the market space is calculated. This sum for a particular product platform should be minimized. The final aim is to get a range of commonality of the design parameters that have minimum variation in the response regardless of changes in the demand or design parameters.

The Market research team at BeamTech has a snapshot of distribution of demand. Based on this distribution, the product platform is designed. Now, according to the augmented method, during the process of designing the platform, there is a slight variation to the demand of every product in the family. Based on these variations in demand with other design variables constant, the corresponding variations in the cost objective are calculated. Hence for every beam considered in the PPCTM, corresponding variance in cost is calculated. The objective is to minimize such variance. The variance in cost is calculated by the following formula:

$$\sigma^2 = \sum (C_i - C)^2 / (n-1)$$

Where C_i is the response calculated after giving variance to demand

C is the response at mean level of demand

n is the number of levels of the noise factor, i.e., demand

Considering the entire market space, the sum of variances of all considered beams in the market space is calculated. This sum for a particular product platform should be minimized. The final aim is to get a range of commonality of the design parameters that have minimum variation in the cost objective regardless of changes in demand. This is achieved by running the PPCTM model with the objective of minimizing the variation in cost.

PPCTM Step 5 - Formulate a Multi-Stage Compromise Decision Support Problem (Manage tradeoff between 'bringing mean on target' and 'minimizing deviation')
(See Figure 3.4)

Step F of managing tradeoff between 'bringing mean on target' and 'minimizing deviation' needs to be integrated into Step 5.

This step is similar to the one in Step 4 (module F) of RCEM (Section 2.2). The evaluation of the two goals of "moving the mean to target" and "minimizing deviation" is completed in Step 4. These are the two goals that need to be satisfied to get a robust design. Under the first goal of "moving the mean to target", there can be a number of objectives that the product platform must achieve. A decision has to be made that would give the best possible combination of the objectives of the product platform, i.e., "moving the mean to target" and "minimizing deviation". The Compromise DSP is used to formulate this problem.

The system goals are formulated for each decision stage of PPCTM. The goal for each stage is based on the individual objective functions of response and variance in response. The values of response and variance are found using the average value of response and summation of variances of response, calculated in each decision stage.

The author's aim behind using cDSP is to find values for the system variables that satisfy the constraints and bounds on the design and achieve as closely as possible two system goals:

1)   moving the mean to target

2)   minimizing the deviation in response

The extent to which each goal is achieved is modeled by the system goals-

$$\text{Response/Ideal Value} + d_i^- - d_i^+ = 1$$

$$\text{Ideal Value/Variance} + d_i^- - d_i^+ = 1$$

These relations are obtained from (Mistree et al., 1993). If objective is to be minimized, goal is formulated as follows: $A_i(x) + d_i^- - d_i^+ = G_i$. The normalized form is

$$\frac{A_i(x)}{G_i} + d_i^- - d_i^+ = 1$$

. If the objective is to be minimized and the goal is non-zero, it is

formulated as follows: $G_i + d_i^- - d_i^+ = A_i(x)$. The normalized form is

$$\frac{G_i}{A_i(x)} + d_i^- - d_i^+ = 1$$ where,

G_i - Goal value for a particular objective i

$A_i(x)$ - Achievement of objective i for a particular design x

In this case, the first goal is to maximize the objective and the second goal is to minimize the objective. A designer would like to achieve the ideal value of 1, but does not expect to achieve it necessarily. Deviation variables (d^- and d^+) indicate the extent to which each goal achieves its target value. The deviation function is a weighted sum of deviation variables for both goals and measures the extent to which both goals are achieved (Seepersad et. al., 2002). The weights for each goal (W_1 and W_2) are used to emphasize achievement of one goal more than another. The deviation function is minimized in the solution process.

$$Z = W_1\, d_1^- + W_2\, d_2^-$$

With the general formulation of the cDSP presented, the details of each stage's specific formulation are presented in the examples. A general solution algorithm is used to solve the formulated cDSP. The solution process of the robust design is illustrated in the examples.

BeamTech has completed formulating the objective functions – minimizing the cost, minimizing the deflection and minimizing the variation in costs. The first two objectives of minimizing the cost and deflection "bring the mean to target" and third objective of minimizing the variation in costs "minimizes deviation". "Bringing the mean to target" and "minimizing deviation" are the two goals that need to be satisfied to get a robust design. A decision has to be made that would give the best possible combination of the objectives of the product. BeamTech formulates Compromise DSP for each decision stage of PPCTM to find values for the system variables that satisfy the constraints and bounds on the design and achieve as closely as possible.

The extent to which each objective is achieved is modeled by the system goals-

$$\text{Ideal Value/Cost} + d_1^- - d_1^+ = 1$$

$$\text{Ideal Value/Variance} + d_2^- - d_2^+ = 1$$

$$\text{Ideal Value/Deflection} + d_3^- - d_3^+ = 1$$

The aim is to achieve the value of the objective function as close to the ideal value as possible. The deviation function is a weighted sum of deviation variables for the three objectives and measures the extent to which both goals are achieved (Seepersad et. al., 2002). The weights for each objective (W_1, W_2 and W_3) are used to emphasize achievement of one goal more than another. The deviation function is minimized in the solution process.

$$Z = W_1 d_1^- + W_2 d_2^- + W_3 d_3^-$$

3.2 STRATEGIC DESIGN OF PRODUCT PLATFORMS USING COMPROMISE DSP

In this section, the steps needed to augment the PPCTM so that strategic design of product platforms is possible, are discussed. As discussed in Section 2.4, strategic design involves taking action in accordance with the future trends well in advance. Chamberlain recognizes that the company should develop forecasts of future market trends to make sure that the money invested in a new product platform or in the development of a new piece of technology continues to pay dividend when the next big market shift occurs a few months or years down the line (Chamberlain, 2002). All of the actions that the company needs to take at the current time to recognize and account for future trends fall under the general heading of an activity that is defined as strategic design. In this section, the author augments the PPCTM with certain steps that helps design strategically. These steps involve exploring potential portfolio expansions and considering its objectives.

BeamTech has forecasted that the present market in which they plan to sell their beams will get saturated in about 2 years. The demand will become lesser as time progresses. Moreover, smaller companies that can compete will further decrease their market share. They would also not want to develop a product line in the same market that will result in cannibalization of demand. So, BeamTech wishes to invest in a product line such that it can be leveraged into neighboring markets after 2 years. Thus BeamTech wants to strategically design their new product platform such that it continues to pay dividend when the next big market shift occurs after 2 years.
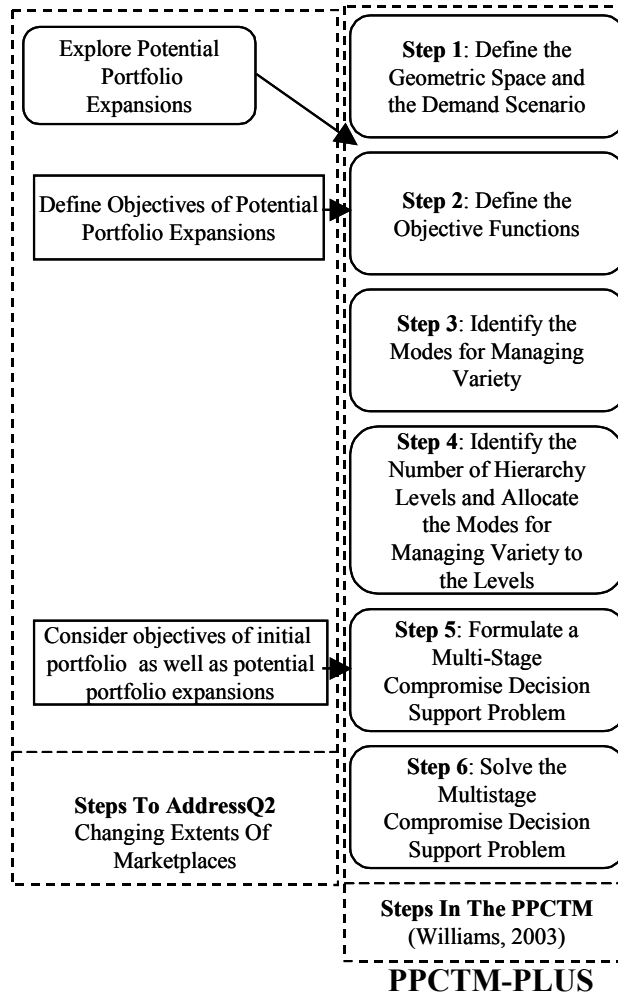
Figure 3.8: Augmentations made to PPCTM-PLUS, for a changing marketplace

Steps to Design Product Platforms for Changing Marketplaces

<u>Step A - Explore potential portfolio expansions</u> (See Figure 3.8)

In this step, the future portfolio expansions are explored. This involves forecasting the market and assigning probabilities to the probable market expansions. The bounds of which product specifications are limited and need to be expanded are studied. The extent to which this product specification needs to be extended is found. The

probability of the demand for that extension is calculated. Hence the result of this step is that a number of potential portfolio expansions are identified, their extent is found and the probability of that situation to arise in the future determined.
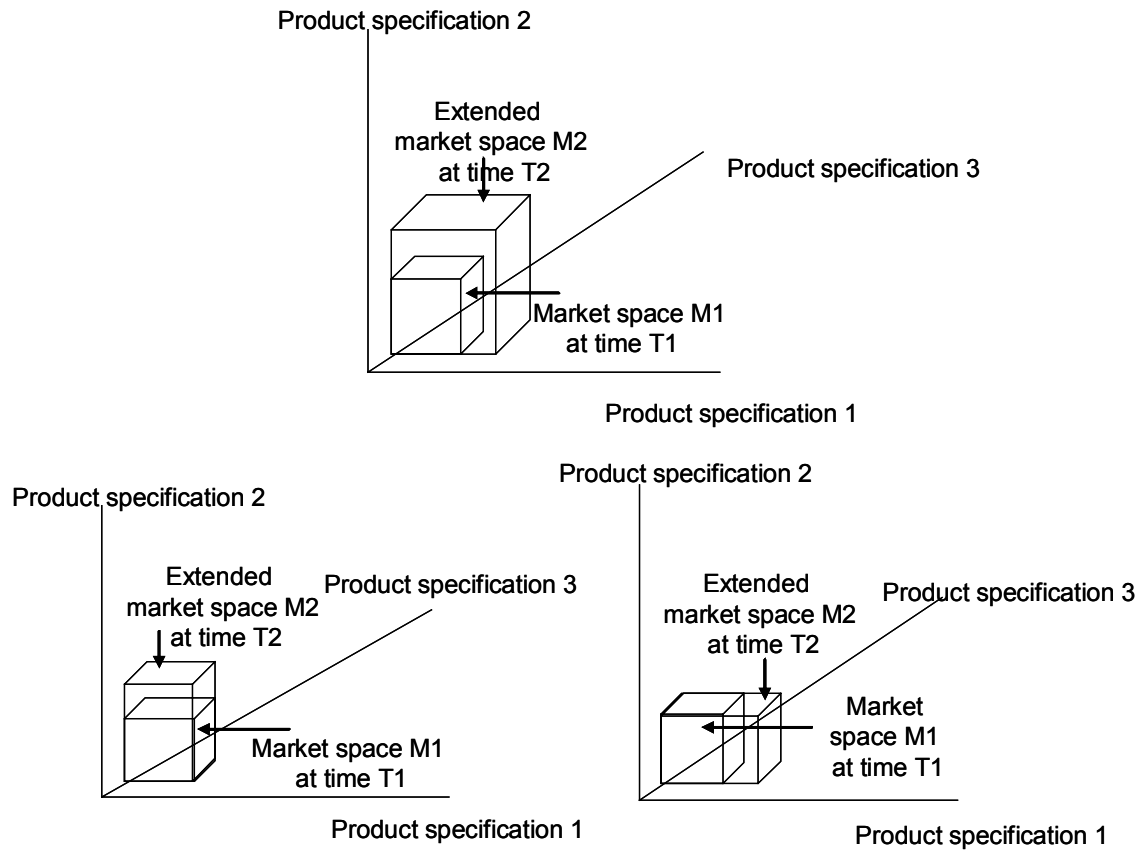
Figure 3.9: Leveraging Product Platforms

In this step, the market research team at BeamTech explores different options for BeamTech to expand its base. BeamTech has three options: expand only the market extent of the load capacity (Figure 3.10a), keeping the extent of length space constant, or expand only the extent of length space (Figure 3.10b), keeping the extent of load capacity constant or expand both length and load capacity market extents (Figure 3.10c). Based on their study, the probabilities are assigned to each of the expansion depending on it to happen. Also, the demand generated in that area needs to be considered.
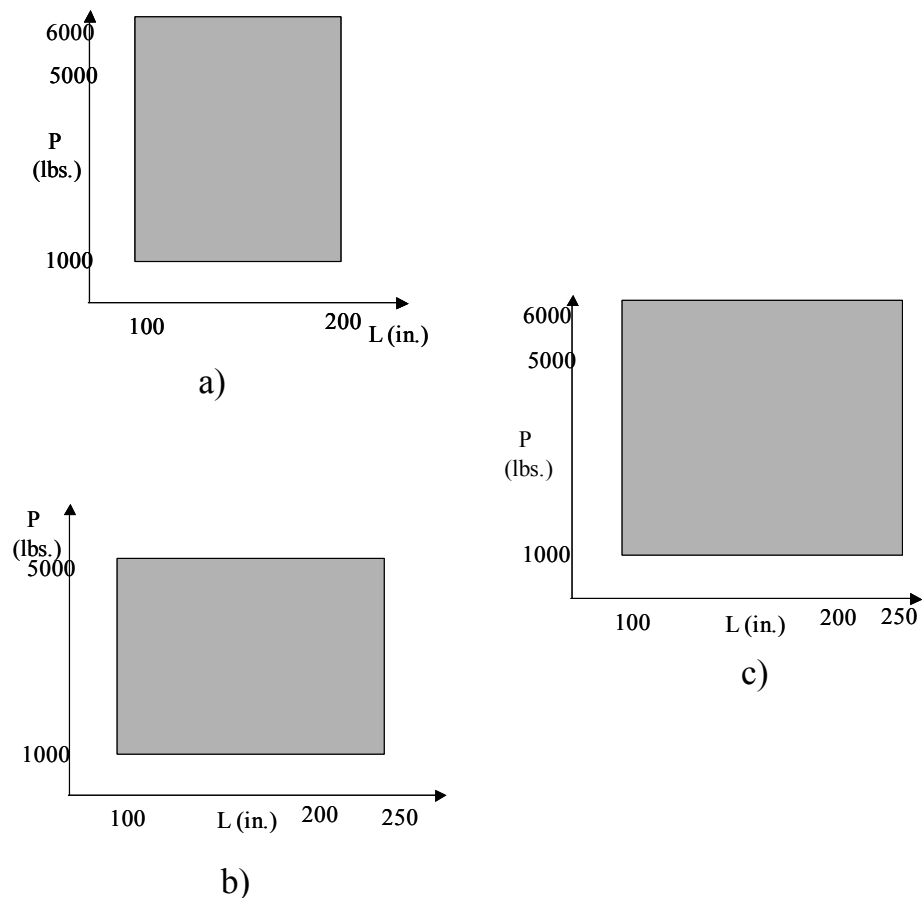
Figure 3.10: Different Possibilities of Expansion of Cantilever Beam Market

100

It is assumed that market research at BeamTech found that there is a probability of 0.2 for the expansion to take place in the vertical direction, 0.4 to take place in the horizontal direction and 0.4 to continue in the existing market.

PPCTM Step 2 - Define the Objective Functions (Define objectives for Potential Portfolio Expansions (See Figure 3.8)

Step C of defining objectives for potential portfolio expansions needs to be integrated into Step 2.

Hernandez and Williams have focused on the goal of achieving the performance objectives, i.e., bring mean on target. This goal includes typical objectives such as minimizing average cost, maximizing average profit, and improving design performance objectives (i.e., strength, mass, etc.). Now, we need to consider the future potential marketplaces. So here apart from the objectives of the initial marketplace, one needs to consider the objectives of the potential marketplaces. The objectives of these potential portfolios of the future may be same or different than the objectives of the initial product portfolio. However, normally the objectives of the future product portfolios will remain the same. Thus in this step, one needs to define the objective functions of the initial product portfolio as well as the future potential portfolios.

It is not necessary that the objectives of BeamTech in the new extension are necessarily the same as the objectives in the old market space. In this step, the objectives in the potential portfolio expansion are defined. For example, in the case of expansion of the extent along the load (vertical direction), BeamTech wishes to add another objective of increasing the life of the beam. Hence, it is seen that a different set of objectives can be defined for the expansions.

PPCTM Step 5 - Formulate a Multi-Stage Compromise Decision Support Problem (Consider objectives of initial portfolio as well as potential portfolio expansions)
(See Figure 3.8)

Step E that considers objectives of initial portfolio as well as potential portfolio expansions needs to be integrated into Step 5.

There are a number of sets of objectives that need to be achieved. Every potential product portfolio will have its own set of objectives. The cDSP is used to determine the ranges of the modes for managing variety that provide the best compromise between these conflicting goals.

Following the tenets of constructal theory, the determination of the range of the modes for managing variety that compose a level of the hierarchy represents one stage in a multi-stage decision. With the order of the use of the modes established, a designer defines a cDSP for each decision stage in this step. The values of decision variables Δr are the ranges of application of each mode. The manner in which the ranges of application of each mode are determined at each stage when one needs to consider the different extents of the product platforms is explained below.

At every stage in the PPCTM, the size or range of the element of that stage is determined. Previously, the size of the element that best achieves the objective is determined. However, now the problem becomes complex. One needs to find the size of the element for different extents of the market. So for every market extent the size of the element in order to maximize their objectives is determined. Depending on the probability of each of these expansions determined in the beginning, the weights of each market extent and hence each objective is determined. This problem to obtain the resultant first stage element is formulated using the compromise DSP.

Consider a case in which there are two possible expansions that are likely in the future. The probabilities of the expansion are p1 and p2. Considering the first stage, the size of the element for each extent of market space is considered. If considered individually, every market space will have its own size. However, we need to find a compromise between the objectives of the product platforms. For instance, if the current or initial market space is only considered, one obtains X1 and Y1 as the size of the element for that space. Similarly now consider potential expansion 1 and 2 individually. For potential expansion 1, since the leveraging takes place only in the horizontal direction, Y1 remains the same, however; a different value of X2 is obtained that achieves the desired objectives. Similarly, for potential expansion 2, since the leveraging takes place only in the vertical direction, X1 remains the same, however; a different value of Y2 is obtained that achieves the desired objectives. Depending on the probabilities of the initial and potential expansions, a compromise is developed between the sizes of the elements. Formulating and solving the compromise DSP, the size of the first stage

element as X and Y is determined. This cDSP manages the tradeoff between the objectives of the current market space and potential expansions for the first stage.
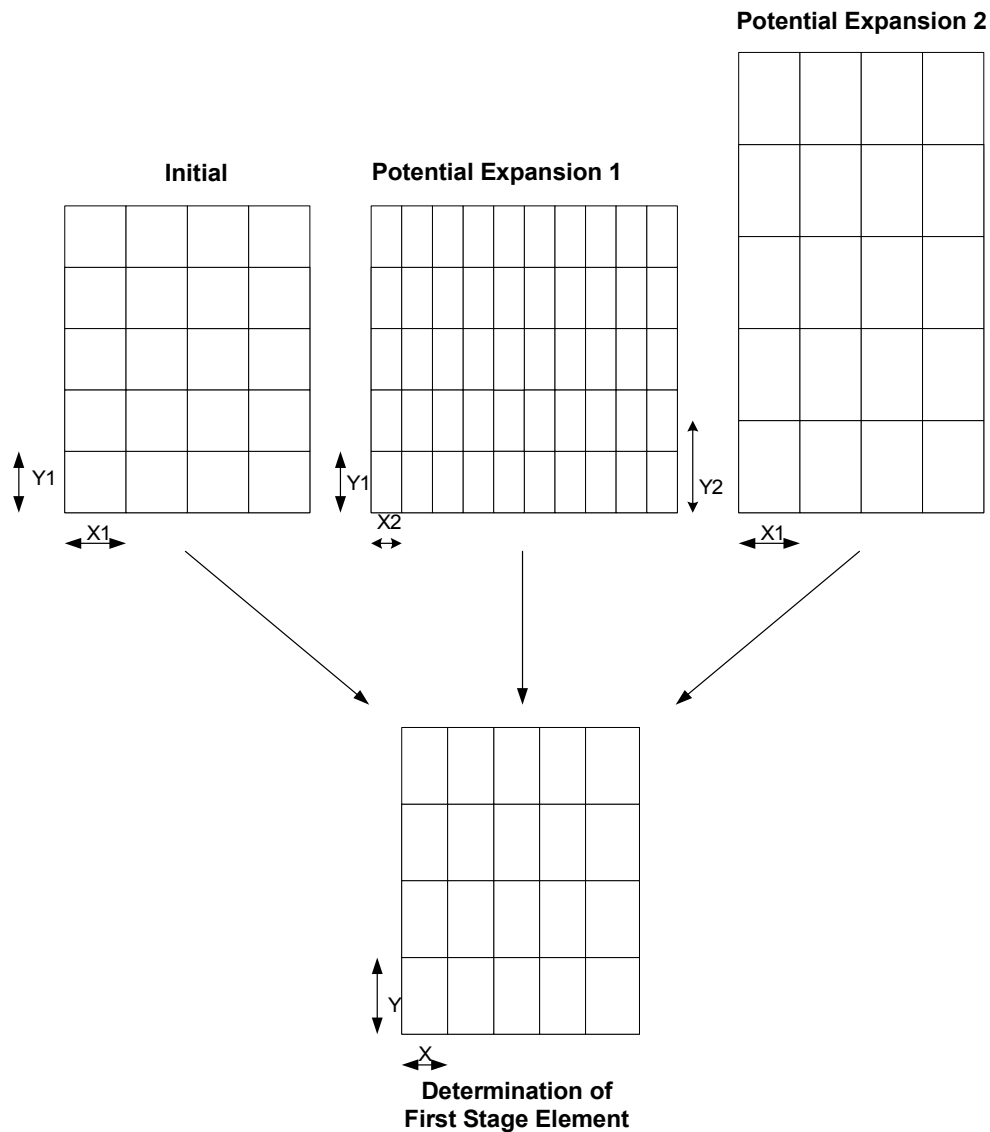
Figure 3.11: First stage element

Now, consider the second stage that involves change of mode of product variety in only one direction. Since in the potential expansion 1 there is leveraging only in the horizontal direction, the size of the second stage element for potential expansion 1 and

the initial/current market space remains same, i.e., P1 and Q1. However, in the potential expansion 2 there is leveraging in the vertical direction, hence the size of the second stage element for potential expansion 2 is different – P2 and Q1. Again, a compromise DSP is formulated and solved to determine the second stage element.

Similarly, for different stages, a compromise DSP is formulated and solved to get the sizes of different elements.
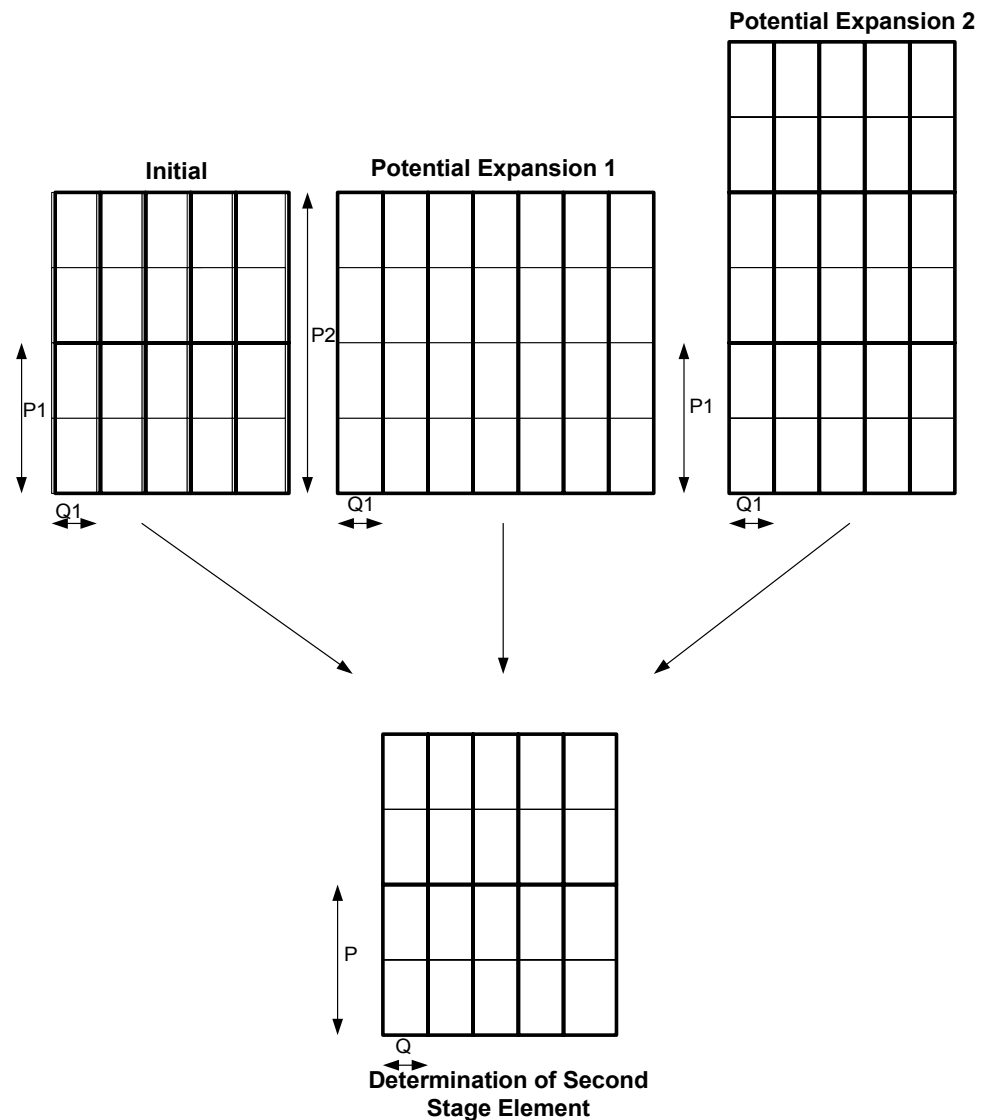
Figure 3.12: Second stage element

The goal for each stage is based on the individual sets of objective functions of each product portfolio. The deviation function is a weighted sum of deviation variables for every objective and measures the extent to which both goals are achieved (Seepersad et. al., 2002). The weights for each goal are used to emphasize achievement of one goal more than another. The aim is to minimize the deviation function developed from the deviation variables and weights. A generalized formulation of the multi-stage cDSP is illustrated in Figure 3.13.

For Each Stage i

Given: The N-dimensional market space $M^N = \{(r_1, r_2, \ldots r_N)\}$

The N-dimensional future potential market space $M^N = \{(r_1, r_2, \ldots r_N)\}$

Probability of the future market scenario

The decision variables of the previous stages $\Delta r(1), \ldots, \Delta r(i-1)$

The modes of managing product variety to be utilized at Stage i

Find: The value of decision variable $x(i) = [\Delta r_1(i), \Delta r_2(i), \Delta r_N(i)]$

The deviation variables, $d_{x,i}^-$ and $d_{x,i}^+$

Satisfy: Bounds: $\Delta r_{j,\min}(i) \le \Delta r_j(i) \le \Delta r_{j,\max}$

Constraints: $\Delta r_j(i) \ge \Delta r_j(i-1)$, $\begin{aligned} d_{x,i}^-, d_{x,i}^+ &\ge 0 \\ d_{x,i}^- \cdot d_{x,i}^+ &= 0 \end{aligned}$

Goals: $G_i + d_{x,i}^- + d_{x,i}^+ = 1$

Minimize: $Zj = \sum W_i x\, (d^-_{x,j} + d^+_{x,j})$

Figure 3.13: Formulation of the Multi-Stage Compromise Decision Support Problem

In the case of BeamTech, the expansions are in the horizontal and vertical directions. So determination of the first stage and second stage elements will be identical to the explanations given above using Figure 3.11 and 3.12. X and Y signify ΔL and ΔP_1 respectively, while P and Q signify ΔL and ΔP_2. So in the first stage, determination of ΔL and ΔP_1 takes place, while in the second stage, determination of ΔP_2 takes place. There is no mode of variation of length in the second stage. Basically, at every stage the ranges of application of the modes are found based on the different objectives of the existing and possible expansions of the market.

3.3 A METHOD FOR DESIGNING PRODUCT PLATFORMS FOR AN UNCERTAIN DISTRIBUTION OF DEMAND, CHANGING DESIGN PARAMETERS AND CHANGING EXTENTS OF MARKETPLACES

A number of augmentations are needed to the existing PPCTM so that it handles uncertain distribution of demand, changing design parameters and changing extents of marketplaces (challenges discussed in Section 1.1.3). The augmented steps have been shown in Sections 3.1 and 3.2. Now these steps are combined into PPCTM-PLUS to present PPCTM-RCM (Robust to Changes in Market) (Figure 3.14).

Step 1: Define the Geometric Space and the Demand Scenario

Step 2: Explore Potential Portfolio Expansions

Step 3: Classify Design Parameters

Step 4: Define the Objective Functions of Initial and Potential Product Portfolios

Step 5: Identify the Modes for Managing Variety

Step 6: Identify the Number of Hierarchy Levels and Allocate the Modes for Managing Variety to the Levels

Step 7: Formulate a Multi-Stage Compromise Decision Support Problem

Step 8: Solve the Multistage Compromise Decision Support Problem

Explore Potential Portfolio Expansions

Step 1: Define the Geometric Space and the Demand Scenario

Classify Design Parameters

Define Objectives of Potential Portfolio Expansions

Step 2: Define the Objective Functions

Development of the Variance Objective Function

Step 3: Identify the Modes for Managing Variety

Step 4: Identify the Number of Hierarchy Levels and Allocate the Modes for Managing Variety to the Levels

Consider objectives of initial portfolio as well as potential portfolio expansions

Step 5: Formulate a Multi-Stage Compromise Decision Support Problem

Manage Tradeoffs Between 'Bringing Mean On Target' and 'Minimizing Deviation'

Step 6: Solve the Multistage Compromise Decision Support Problem

Steps To Address Changing Extents Of Marketplaces

Steps To Address Uncertain Distribution Of Demand and Changing Design Parameters

Steps In The PPCTM (Williams, 2003)

Augmentations To The PPCTM

Figure 3.14: Augmented PPCTM

108

3.3.1 Step 1: Define the Geometric Space and the Demand Scenario (See Figure 3.14)

In the first step of the PPCTM the space of customization is defined. The *space of customization* is the set of all feasible combinations of values of product specifications that a manufacturing enterprise is willing to satisfy (Hernandez et al., 2002). Consider that there are N independent product requirements $r_1, r_2,...,r_N$ identified that characterize the customer demands on a product. These requirements help to define the N-dimensional space of customization $M_n = \{(r_1, r_2, ..., r_N)\}$. A space of customization definition involves the following components:

- identifying which parameters of the product should be varied depending on the needs of the customer,

- the range of variety that is needed to be offered for each parameter,

- the customer demand in the space of customization,

- the possible variability in demand in the future.

Each dimension of the geometric space represents one of the product parameters in which variety will be offered. The bounds of each dimension of the geometric space are determined by the range of each varied parameter. The PPCTM is capable of handling non-uniform demand scenario (Williams, 2004).

3.3.2 Step 2: Explore Potential Portfolio Expansions (See Figure 3.14)

In this step, the future portfolio expansions are explored. This involves forecasting the market and assigning probabilities to the probable market expansions. The bounds of which product specifications are limited and need to be expanded are studied. For example, in the topmost part of Figure 3.15, the market (product specifications requirements) are expanded in all the directions while in the lower part expansion takes place in only one direction. The extent to which this product specification needs to be extended is found. The probability of the demand for that extension is calculated. Hence the result of this step is that a number of potential portfolio expansions are identified, their extent is found and the probability of that situation to arise in the future estimated.
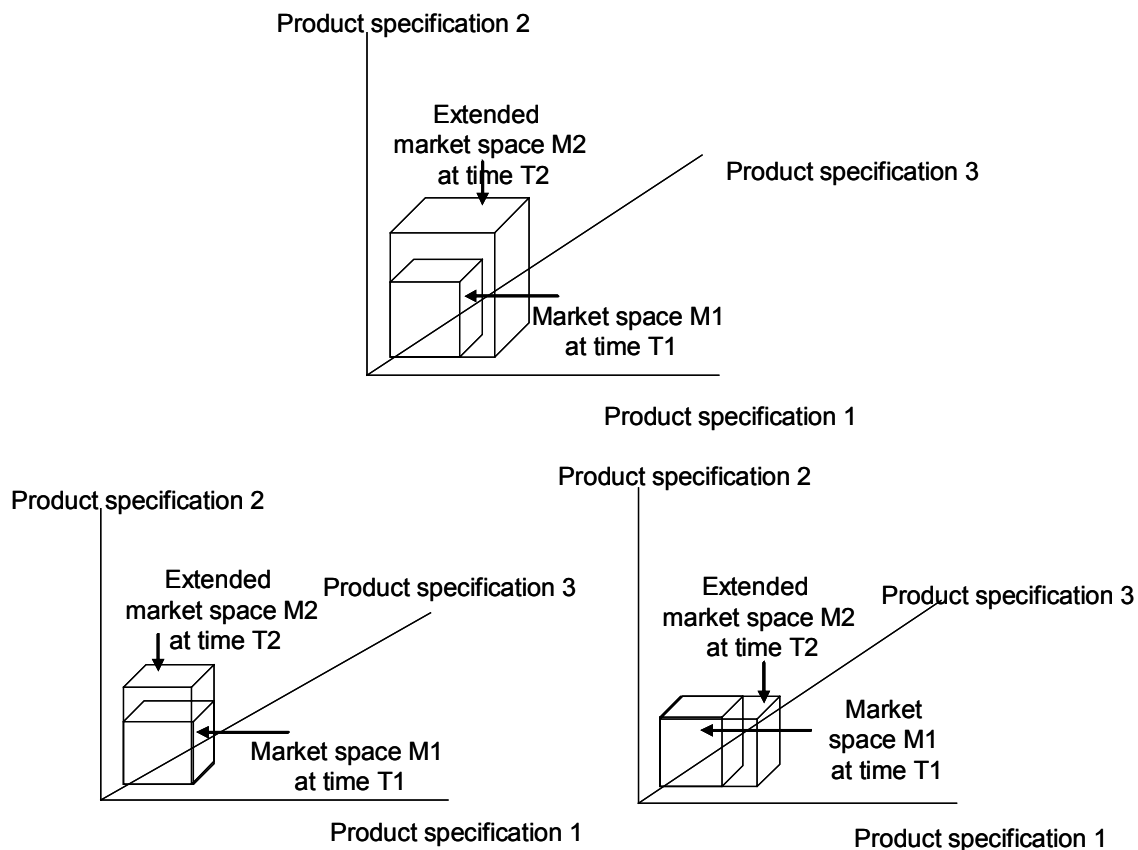
Figure 3.15: Leveraging Product Platforms

3.3.3 Step 3: Classify Design Parameters (See Figure 3.14)

In this step, the initial concept exploration space is defined and the problem is formulated as a robust design. Design variables are grouped as either control factors or noise factors. Ranges of values are specified. The responses, i.e., objective functions that need to be studied are also identified.
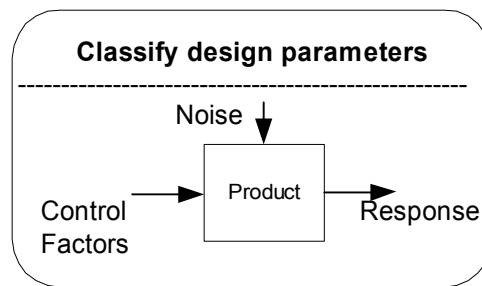
Figure 3.16: Parameter Diagram

As discussed in Section 3.1.1, the method facilitates handling both the types of robust design problems: Type 1 and Type 2. In the method, Type 1 problem involves demand as a noise factor, the product specifications as the control factors and the response as the objectives of that platform. The Type 2 problem involves deviation of the design parameters that are the control factors.

Thus, in this step the classification of different terms in the product platform is done.

3.3.4 Step 4: Define the Objective Functions for Initial and Potential Product Portfolios (See Figure 3.14)

The original PPCTM developed in (Hernandez et al., 2001) focused on the goal of achieving the performance objectives, i.e., bringing mean on target. Williams in (Williams, 2003) infused the utility based compromise Decision Support Problem to handle multiple objectives in the PPCTM. An objective function is formulated by a discretized analysis of the space using a summation equation (Equation 3.2), or through a continuous analysis of the space using an integral (Equation 3.3).

$$o = \sum_{i=r_{1,min}}^{r_{1,max}} \sum_{j=r_{2,min}}^{r_{2,max}} ... \sum_{k=r_{N,min}}^{r_{N,max}} O(r_{1,i}, r_{2,j}, ... r_{N,k}) \qquad [3.2]$$

$$o = \int_{r_{1,min}}^{r_{1,max}} \int_{r_{2,min}}^{r_{2,max}} ... \int_{r_{N,min}}^{r_{N,max}} O(r_1, r_2, ... r_N) dr_1 dr_2 ... dr_N \qquad [3.3]$$

where the *min* and *max* subscripts refer to the upper and lower bounds, respectively, of each dimension of the space.

Changing demand and design parameters - Development of the variance objective function:

As discussed previously in Section 3.1.1, the goal of "moving the mean to target" has already been presented in (Williams, 2003). It is suggested by the author that the development of the other goal of "minimizing variation" could also be achieved by using the PPCTM.

Here, the calculation of the variance objective function for the Type 1 design problem to handle changing demand is presented. The changes to be made in the variance

objective function for the Type 2 design problem to handle changing design parameters are discussed after this explanation.

For every product that is considered, some amount of noise is provided. This noise represents the uncertainty in distribution of demand. The variation in response due to the noise is noted. From these values of responses, the variance in response is calculated. The amount of noise that is provided can have any value.

So, for a product (keeping all other design variables constant) with certain values of noise, the corresponding variance in response is calculated. Hence for every product considered in the PPCTM, corresponding variance in response is calculated. The objective is to have minimum such variance. The variance in response is calculated by the following formula:

$$\sigma^2 = \sum (R_i - R)^2 / (n-1) \qquad [3.4]$$

Where R_i is the response calculated after giving noise N_i to demand

R is the response at mean level of demand

n is the number of levels of the noise factor

Considering the entire market space, the sum of variances of all considered products in the market space is calculated. This sum for a particular product platform should be minimized. The final aim is to get a range of commonality of the design parameters that have minimum variation in the response regardless of changes in demand. This is achieved by running the PPCTM model with the objective of minimizing the variation in objective.

Similar analysis needs to be done to calculate the variance objective function for the Type 2 design problem involving changing design parameters. Here, the deviation is provided in one or more of the design parameters (control factors). For every product in the platform, the effect of the deviation in the design parameter on the response is determined. Considering the entire market space, the sum of variances of all considered products in the market space is calculated. This sum for a particular product platform should be minimized.

Changing marketplaces – Initial and potential product portfolios:

The goals discussed in the previous section includes typical objectives such as minimizing average cost, maximizing average profit, and improving design performance objectives (i.e., strength, mass, etc.) for the current marketplace. Now, we need to also consider the future potential marketplaces. So here apart from the objectives of the initial marketplace, one needs to consider the objectives of the potential marketplaces. The objectives of these potential portfolios of the future may be same or different than the objectives of the initial product portfolio. However, normally the objectives of the future product portfolios will remain the same. Thus in this step, one needs to define the objective functions of the initial product portfolio as well as the future potential portfolios.

3.3.5 Step 5: Identify the Modes for Managing Variety (See Figure 3.14)

There is no difference between this step and the original formulation of this step in (Williams et al., 2003) and (Hernanadez et al., 2001). The modes of managing variety represent the ways in which one can provide variety to the customer in the market space. These are the linking mechanism between the product variants that compose the product family. Some modes for managing product variety include dimensional commonalization, component commonalization, modularity, and standardization. There is no defined way of carrying out this step. This step instead involves brainstorming and decision-making between design and manufacturing.

3.3.6 Step 6: Identify the Number of Hierarchy Levels and Allocate the Modes of Managing Variety to the Levels (See Figure 3.14)

Now that the space of customization and the objectives of the product platform are defined, one needs to identify the modes for managing product variety. This step involves the consideration of how a product will be varied in order to satisfy the customized specifications of the space of customization. The number of hierarchy levels can be equal to or less than the number of modes of managing variety. A hierarchy level can have one or more modes of managing variety. These modes of managing product variety are the linking mechanism between the product variants that compose the product family. There is no rigid structure to the formulation of these modes; this step instead involves brainstorming and decision-making between design and manufacturing. The allocation of the modes to appropriate hierarchy levels is done by achieving the smallest variations in the varied design parameters at the lower levels of the hierarchy.

3.3.7 Step 7: Formulate a Multi-Stage Compromise Decision Support Problem
(See Figure 3.14)

According to the principles of constructal theory, each stage in a multi-stage decision represents a level of the hierarchy and each level determines the range of the modes for managing variety. This step is comprised of defining a cDSP at each decision level.

Each decision at every stage involves finding out the range of commonality of each mode for managing variety, $\Delta r(i)$. In other words, this involves determining the size and shape of each subspace. For a problem with N varied parameters, the decision variables for any stage i, are

$$\Delta r(i) = [\Delta r_1(i), \Delta r_2(i), ..., \Delta r_N(i)] \qquad [3.4]$$

In the formulation of each cDSP, there exists a constraint on the selection of the range of commonality for each mode of managing variety so that a hierarchic design is achieved:

$$\Delta r_j(i+1) \geq \Delta r_j(i) \qquad [3.14]$$

The cDSP helps to find values for the system variables that satisfy the constraints and bounds on the design and achieve as closely as possible the system goals. A generalized formulation of the multi-stage cDSP is illustrated in Figure 3.17.

For Each Stage i

Given: The N-dimensional market space $M^N = \{(r_1, r_2, \dots r_N)\}$

The N-dimensional future potential market space $M^N = \{(r_1, r_2, \dots r_N)\}$

Probability of the future market scenario

The decision variables of the previous stages $\Delta r(1), \dots, \Delta r(i-1)$

The modes of managing product variety to be utilized at Stage i

Find: The value of decision variable $x(i) = [\Delta r_1(i), \Delta r_2(i), \Delta r_N(i)]$

The deviation variables, $d_{x,i}^{-}$ and $d_{x,i}^{+}$

Satisfy: Bounds: $\Delta r_{j,\min}(i) \leq \Delta r_j(i) \leq \Delta r_{j,\max}$

Constraints: $\Delta r_j(i) \geq \Delta r_j(i-1)$

$$d_{x,i}^{-}, d_{x,i}^{+} \geq 0$$
$$d_{x,i}^{-} \bullet d_{x,i}^{+} = 0$$

Goals: $G_i + d_{x,i}^{-} - d_{x,i}^{+} = 1$

Minimize: $Zj = \sum W_i x (d^{-}_{x,j} + d^{+}_{x,j})$

Figure 3.17: Formulation of the Multi-Stage Compromise Decision Support Problem

Changing distributions of demand and design parameters

As stated in Section 3.1.1, this step is similar to the one in Step 4 (module F) of RCEM (Section 2.2). The evaluation of the two goals of "moving the mean to target" and "minimizing deviation" is discussed in Step 4 of this section. These are the two goals that need to be satisfied to get a robust design. Under the first goal of "moving the mean to

117

target", there can be a few objectives that the product platform must achieve. A decision has to be made that would give the best possible combination of the objectives of the product platform, i.e., "moving the mean to target" and "minimizing deviation". The Compromise DSP is used to formulate this problem.

The aim of the cDSP is to find values for the system variables that satisfy the constraints and bounds on the design and achieve as closely as possible two system goals:

1) moving the mean to target

2) minimizing the deviation in response

The extent to which each goal is achieved is modeled by the system goals-

$$\text{Response/Ideal Value} + d_i^- - d_i^+ = 1$$

$$\text{Ideal Value/ Variance} + d_i^- - d_i^+ = 1$$

The aim is to maximize the value of each individual objective function (i.e., reach the RHS, 1). A designer would like to achieve the ideal value of 1, but does not expect to achieve it necessarily. Deviation variables (d^- and d^+) indicate the extent to which each goal achieves its target value. The deviation function is a weighted sum of deviation variables for both goals and measures the extent to which both goals are achieved (Seepersad et. al., 2002). The weights for each goal (W_1 and W_2) are used to emphasize achievement of one goal more than another. The deviation function is minimized in the solution process.

$$Z = W_1 \, d_1^- + W_2 \, d_2^-$$

There are a number of sets of objectives that need to be achieved in a changing environment. Every potential product portfolio will have its own set of objectives. The cDSP is used to determine the ranges of the modes for managing variety that provide the best compromise between these conflicting goals.

The manner in which the ranges of application of each mode are determined at each stage, when one needs to consider the different extents of the product platforms is explained below.

At every stage, in the PPCTM the size or range of the element of that stage is determined. Previously, the size of the element that best achieves the objective is determined. However, now the problem becomes complex. One needs to find the size of the element for different extents of the market. So for every market extent the size of the element in order to maximize their objectives is determined. Depending on the probability of each of these expansions determined in the beginning, the weights of each market extent and hence each objective is determined. This problem to obtain the resultant first stage element is formulated using the compromise DSP as shown in Section 3.1.1.

3.3.8 Step 8: Solve the Multi-Stage Compromise Decision Support Problem (See Figure 3.14)

The final step of the augmented PPCTM is solving the multi-stage cDSP formulated in the previous step. Solution of the different stages of the cDSP helps to obtain the values of the ranges of each mode for managing product variety that achieve

closest to the ideal value of the objectives. Although the solution method proposed in the formulation of the original PPCTM is the use of dynamic programming to approximate the relationship between each decision stage (Hernandez, 2001), Hernandez has moved to a more generic description of his sixth step, "Solve the Multi-Stage Optimization Problem" in his recent publications (Hernandez et al., 2002). The difficulties in dynamic programming due to the exhaustive search of combinations of combinations of design parameters and their influence on the objectives are noted down in (Williams, 2003). Following this trend, this step involves the implementation of "an appropriate solution algorithm" for the determination of the main decision variables of the multi-stage cDSP.

This solution method (modified from Williams, 2003) involves iterating through values of the modes of managing variety (Δr) establishing the dimensions of the sub-spaces, commonalizing the design parameters across each sub-space, evaluating the objective functions, and calculating and comparing the resulting overall utility of each iteration.
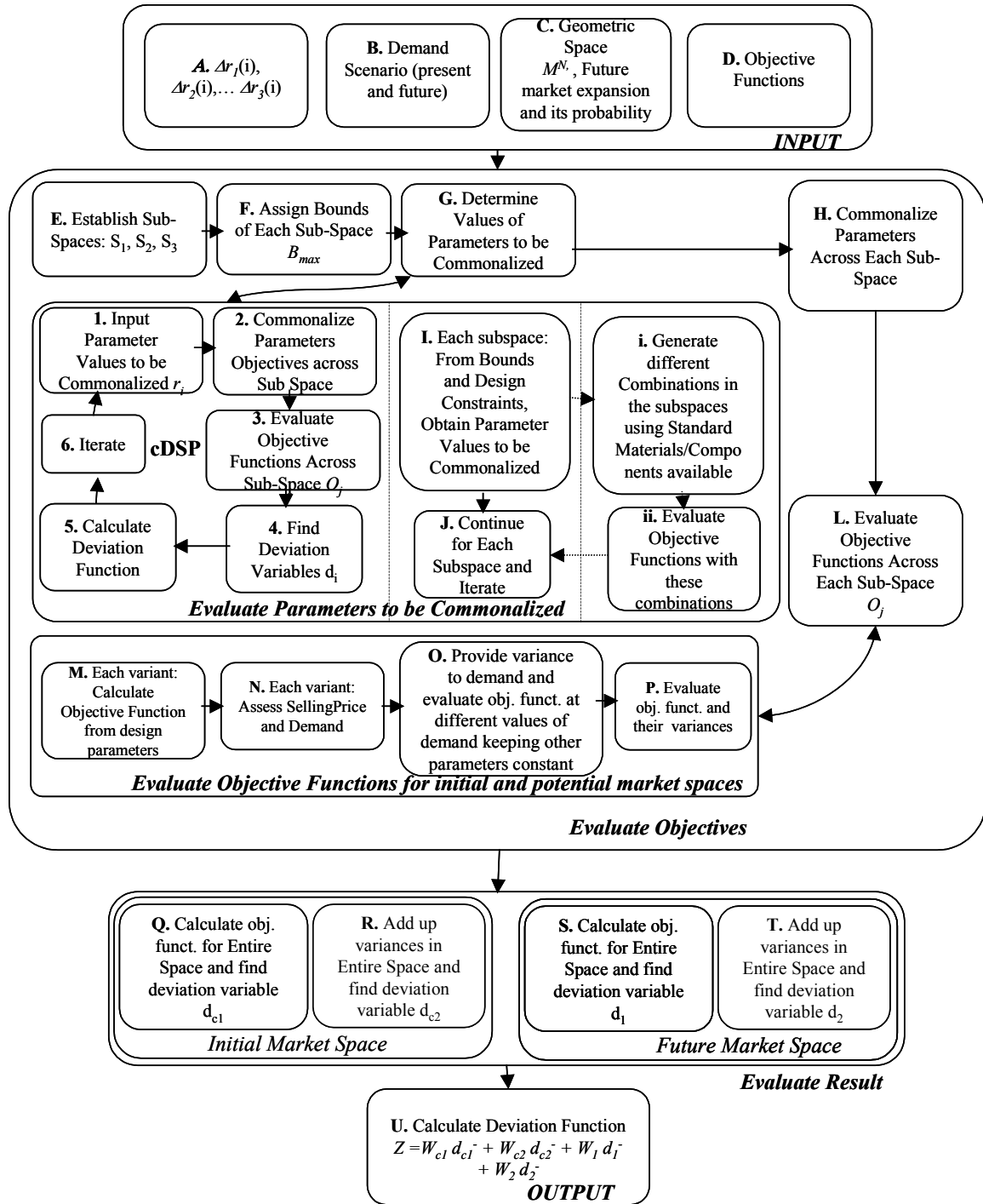
Figure 3.18: Solution Algorithm (Modified from Williams, 2003)

121

-        A particular combination of sub-space element sizes is taken as input **(A)**. The input also includes demand scenario **(B)** (present and future) (Step 1), current geometric space (market space) (Step 1) and future potential expansion of the market with its probability **(C)** (Step 2) and the objective functions **(D)** (Step 4).

-        From the sizes of the sub-space elements **(E)**, the bounds of each sub-space are established **(F)**. The values of parameters to be commonalized are determined from the bounds of the sub-space **(G)**. This involves calculation of the parameters from their respective modes **(I, J)**. This step of calculation of the parameters can also be done by formulating cDSPs and finding out the parameters with the goal of achieving the objectives **(1, 2, 3, 4, 5, 6)**. There is a need to generate different combinations of the standard materials or components if available for a particular mode **(i, ii)**. Then objective functions are determined based on this combination.

-        These parameters are then commonalized across the respective sub-spaces **(H)**. The objective function is determined for every variant considering the parameter values across each sub space **(L)**.

-        For calculation of the variance in response, the first step involves calculation of the response of the system for the particular parameter values **(M)** and assessing the selling price **(N)** and demand for each variant. Then variance is provided in the form of noise or deviation to the control factors. The response is then calculated while keeping other parameters constant. From the responses obtained due to different values of the deviation in control factors or noise, the

variance in response is calculated **(O)**. Approximately, ±20% noise is provided to the values of demand or deviation in control factors.

- After obtaining the variance in response for each variant, it is added up to get the total variance. Similarly, the response of each variant is summed and averaged across the space **(P)**. The objectives are found for the initial as well as potential expansion of the market.

- The deviation variables of each of the objectives are obtained **(Q, R, S, T)**. Depending on the probability of the future expansion, the weights for each of the objectives of the cDSP are decided. The deviation variables and the weights of profit and variance for the initial and potential market spaces help to obtain the deviation function as the output of this algorithm **(U)**.

- The value of the deviation function obtained from the deviation variables is returned from the analysis and is compared with previous output. After each feasible combination of ranges of the modes for managing variety is explored, the input that provides the smallest deviation from the goal is selected.

This is a general algorithm by which the decision variables for the multi-stage cDSP formulated in Step 7 is determined. Using this algorithm, any solution technique can be used to get the decision variables. Although this solution method is not the most efficient, it is rigorous and is capable of providing empirical structural validity to the research goals of this work. The research goal of this thesis is not to explore different solution algorithms or find the best solution technique.

In recent publications of Hernandez (Hernandez et al., 2003), exhaustive search is used for multi-stage, simple problems. In his thesis, Williams has analyzed different optimization techniques along with the exhaustive search. It has been found that not only exhaustive search technique gives better results than other methods, but the computational intensiveness is also manageable. Hence, the author follows the trend to use exhaustive search technique in this thesis comparing it with optimization techniques in one of the examples.

This augmented PPCTM facilitates design of products that are robust to uncertain distribution of demand, changing design parameters and extents of marketplaces.

3.4 A LOOK BACK AND A LOOK AHEAD

The author's objective in this thesis is to augment the PPCTM to enable it to handle:

- uncertain distribution of demand

- changing design parameters

- changing extents of marketplaces

In Chapter 2, the different constructs on which the augmentations of the PPCTM are proposed are discussed. The constructs discussed are:

- Robust Concept Exploration Method

- Compromise DSP

- Strategic Design

In this chapter, the different augmentations are proposed using the constructs in Chapter 2. The augmented PPCTM has also been presented that overcomes the limitations stated in Section 2.4.

The first two sections deal with proposing the steps that are needed to handle the above mentioned augmentations. In the first section, the author addresses the first two augmentations – uncertain distribution of demand and changing design parameters. This section helps to integrate aspects of the RCEM into product platform design to handle robustness. The steps needed for Type 1 and Type 2 robust design problems have been discussed in this section.

The author's objective in the second section is to address the third augmentation – changing extents of marketplaces. This section helps to propose steps by which compromise DSP is used to strategically design product platforms. This involves exploring new portfolio expansions and designing product platforms considering current as well as future product portfolio objectives.

In the third section, the augmented PPCTM has been presented. The augmented eight-step method facilitates handling of the above-mentioned three objectives of this thesis. Thus it enables a designer to design product platforms that are robust to changes in demand, design parameters and extents of marketplaces.

Looking at the theoretical structural validity of the thesis, it started in Chapter 2 where the different underlying constructs that are going to be used were discussed. In this chapter, the theoretical structural validity is established by showing how the constructs are combined together to achieve the objectives of this thesis. The theoretical structural

validity is established in this chapter by considering the author's three objectives in the thesis:

- The steps making the product platform robust to changing demand and design parameters involve using the basic principles of robust design. The aspects from RCEM remain intact through the infusion and hence the resulting augmentation is sound. The incorporation of multiple objectives has already been established in William 2003 and hence PPCTM remains structurally sound after the incorporation of robustness.

- Compromise DSP has been used to strategically design product platforms in the thesis. The use of compromise DSP for product platforms design has been done in (Seepersad, 2002). The combination of the use of compromise DSP for strategic design and PPCTM for design of platforms has been presented in this chapter without violating any of the individual formulations.

Looking back, the challenges that need to be faced in design of product platforms for customization has been discussed in Chapter 1. The constructs that are used to overcome some of the limitations identified in Chapter 1 has been presented in Chapter 2. Using those constructs, the PPCTM has been augmented to address the key issues and the resulting eight step method is presented in this chapter. The application of this method for empirical structural and empirical performance, are presented in Chapters 4 and 5. In Chapter 4, the method is applied to the design of customizable pressure vessels and hand exercisers to show the method's ability to handle uncertain demand and changing design

parameters respectively. In Chapter 5, the augmented method is used in the design of a

line of pressure vessels to deal with the changing extents of the market.

CHAPTER IV

DESIGN OF A FAMILY OF CUSTOMIZABLE PRESSURE VESSELS AND HAND EXERCISERS TO DEAL WITH UNCERTAIN DEMAND AND CHANGING DESIGN PARAMETERS

In Chapter 2, the underlying constructs that are going to be used in this thesis are presented. In Chapter 3, the augmented PPCTM is established using the constructs discussed in Chapter 2. The theoretical structural validity has been established in this thesis till now. This chapter and Chapter 5 apply this method to examples to provide empirical and theoretical performance validity of the work.

The main focus of the author in this chapter is to answer the first section of the primary research question:

Q1.1: *"How should the Product Platform Constructal Theory Method be augmented in order to cope with the uncertainty and inherent changes in demand?"*

Q1.2: *"How should the Product Platform Constructal Theory Method be augmented in order to cope with the changes in design parameters?"*

Hypothesis 1: The infusion of concepts of robustness into the Product Platform Constructal Theory Method enables to design platforms that are unaffected by changes in demand and design parameters.

This chapter is comprised of two example problems – pressure vessel and hand exerciser. The pressure vessel problem helps to address the Q1.1, which deals with the uncertainty and inherent changes in demand. The hand exerciser problem helps to address the Q1.2, which deals with the changes in design parameters. The concepts of robustness, as explained in Chapter 2 and 3, are used to design platforms that are unaffected by such changes. The position of this chapter in the thesis is seen in Figure 4.1 and Figure 4.2.
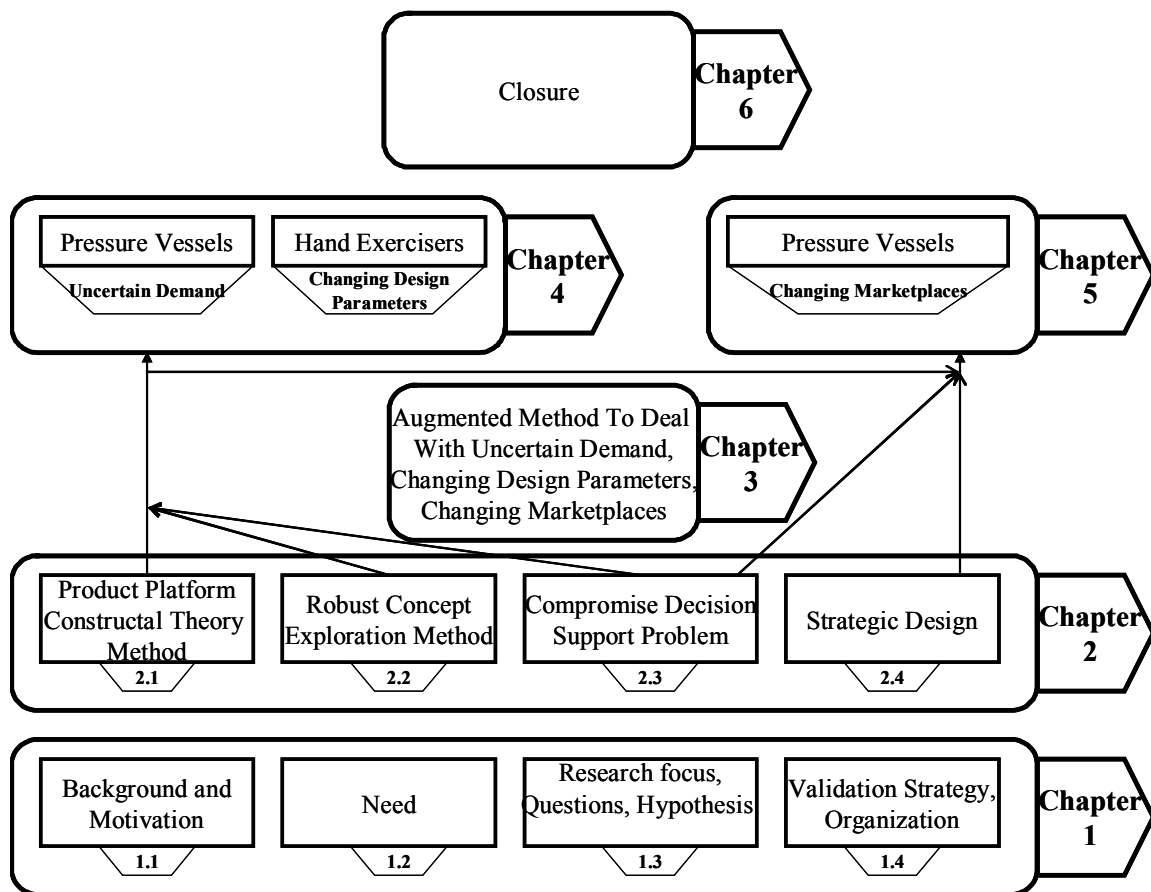
Figure 4.1: Overview of Chapters

	Relevance	Hypotheses
Chapter 1 Designing Product Families for Mass Customization	• Introduction, Motivation and Frame of Reference • Research Questions and Hypotheses • Organization of the thesis	Introduce
Chapter 2 The Theoretical Foundation	• Theoretical foundation of PPCTM • Detailed description of Hernandez's PPCTM • Description of constructs of potential augmentations • Theoretical structural validation	Elaborate
Chapter 3 A Method for Designing Product Platforms to Deal with Uncertain Demands, Changing Design Parameters and Changing Marketplaces	• Infuse concepts of robustness into product platform design • Use compromise DSP for strategic product platform design • Description of the method • Theoretical structural validation	Method for Implementation of Hypotheses
Chapter 4 Customized Product Platform Design in an Uncertain Environment: Pressure Vessels and Hand Exercisers	• Demonstrate that the proposed augmentation is able to deal with changes in demand and design parameters. • Empirical Structural and Performance Validity	Verify H1
Chapter 5 Customized Product Platform Design for a Changing Marketplace: Pressure Vessels	• Demonstrate that the use of compromise DSP is an effective way of strategically designing product platforms for a changing marketplace. • Empirical Structural and Performance Validity	Verify H2
Chapter 6 A Look Back and A Look Ahead	• Summarize and critically review research findings • Contributions and limitations • Recommend future work • Theoretical performance validation	Summarize

Figure 4.2: Thesis Roadmap

In Section 4.1, the pressure vessel example is presented while in Section 4.2 the hand exerciser example is presented. Each part has five sections in it. The first section is comprised of the problem statement. Then the modeling of the example is presented. The

130

third section deals with the application of the augmented method to the example problem. The fourth and fifth sections present the results and the analysis and validity of the results. This chapter helps to present a method that designs product platforms robust to changes in demand and design parameters.

4.1 ROBUST DESIGN OF A FAMILY OF CUSTOMIZABLE PRESSURE VESSELS TO DEAL WITH UNCERTAIN DEMAND (TYPE I ROBUST DESIGN PROBLEM)

This example problem of designing a family of customizable pressure vessels has been presented in previous applications of the PPCTM (Hernandez, 2001; Carone et al., 2003; Williams, 2003; Hernandez et al., 2003). As such, it is an appropriate example for the validation of the augmentations provided in this work since comparison of results is made. Through this example problem, the section of the method to handle uncertain demand is tested.

4.1.1 Problem Statement – Uncertain Demand

A manufacturer of pressure vessels wishes to customize the vessels to gain more customers and has a competitive advantage over other leading manufacturers. As is typical of markets of customizable products, the demand is highly fragmented and uncertain. As such, the manufacturer wishes to develop a product family that can efficiently offer customizable pressure vessels in a market of uncertain demand.

For this example, it is assumed that the conceptual design phase and the beginning of the embodiment phase of the pressure vessel have already been completed. Each

vessel consists of a cylindrical container capped on both ends by hemispherical heads. A schematic of the basic design is presented in Figure 4.3.
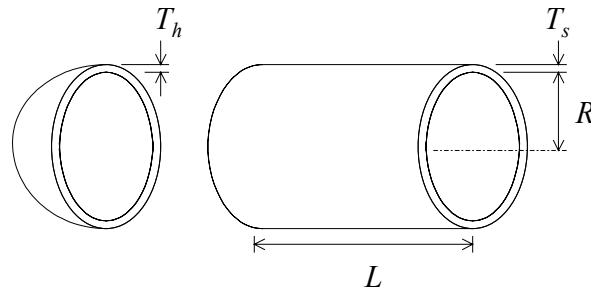
Figure 4.3: Schematic of Pressure Vessel Embodiment (Hernandez, 2001)

The manufacturer wants to specify for each possible combination of pressure and volume the following design variables: length (L), radius (R), and the head and shell thickness (T_h and T_s). The manufacturer wishes to offer vessels that range in volume from 10 to 30 m^3, and in pressure from 10 to 30 MPa. The manufacturer's aim is to have a robust product platform that will give maximum average profit despite changing product demand.

Some of the assumptions in this example are noted down below:

- The development of the product platform is by scaling product components and not by changing functionality of the product.

- In this chapter, the extents of the marketplace are assumed to be fixed. There is no extension of the market in future. However, in Chapter 5, the future extensions of markets are also considered.

- Due to the customization provided in the vessels, it is assumed that the resulting product family will be of high demand and therefore profitable.

4.1.2 Modeling of Pressure Vessel Product

The pressure vessels are produced from carbon steel ASME SA 203 grade B. Sheets of this material are available in thicknesses ranging between 6.35mm and 76.2mm. Available equipment limits the maximum radius to 1.5m and the maximum length to 7m.

The design of the vessels must satisfy the following constraints on the minimum allowable thickness of the shell and for a given pressure (Bednar, 1986):

$$T_s \geq \left(\frac{P}{\sigma_y - 0.6P} \right) R \qquad\qquad [4.1]$$

$$T_h \geq \left(\frac{P}{2\sigma_y - 0.2P} \right) R \qquad\qquad [4.2]$$

where P is the pressure, σ_y is the yield strength of the material (1077 MPa).

In order to assess the profit of the product platform, a means of modeling the selling price and the cost of the vessels is needed. The analysis of the cost is similar to that used by Hernandez in his original application of the PPCTM to the pressure vessel example (Hernandez, 2001) and (Williams, 2003) in his augmentation. The total cost of the manufacturing of the pressure vessel (ignoring all labor and plant costs) includes four components: material cost, welding cost, order cost, and equipment cost.

The material cost is determined by the amount of material that is going to be needed to build the vessel. This is in turn the amount of material purchased. This cost is comprised of two parts - the cost of the material used in each vessel and the cost of the material wasted by cutting the raw steel plates to the required dimensions. The cost of material ($C_{material}$) is therefore given by:

$$C_{material} = 2\pi\rho(C_s R T_s L + C_h R^2 T_h) + C_{waste} \tag{4.3}$$

where ρ is the density of the material (7800 kg/m^3), C_s is the cost per kg of processed

shell steel ($0.8 per kg), and C_h is the cost per kg of forged steel for the head ($2 per kg).

The cost of wasted material (C_{waste}) is given by:

$$C_{waste} = 2\pi\rho C_p T_s R(L_o - L) \tag{4.4}$$

where L_o and C_p are the length and cost per kg ($0.3 per kg), respectively of a raw steel

plate.

All welds in the manufacturing process are single-welded butt joints with a backing

strip. The welding cost (C_{weld}) is composed of the cost of the longitudinal weld and the

cost of the circumferential weld. The longitudinal welding cost ($C_{longweld}$) is given as:

$$C_{longweld} = V_l \rho C_w \tag{4.5}$$

where the volume of the welding material, V_l is given by

$$V_l = 2\pi \left(\frac{T_s}{\cos 30°}\right)^2 \left(\frac{60}{360}\right) L = \frac{4}{9}\pi T_s^2 L \tag{4.6}$$

and C_w is the approximate cost of welding material ($15/kg hand welded).

The circumferential welding cost ($C_{circweld}$)is given as:

$$C_{circweld} = V_s \rho C_w \tag{4.7}$$

where the volume of the welding material, V_s, is given by

$$V_s = 4\pi^2 \left(\frac{T_s}{\cos 30°}\right)^2 \left(\frac{60}{360}\right) R = \frac{8}{9}\pi^2 T_s^2 R \tag{4.8}$$

The total welding cost becomes

$$C_{weld} = 2\pi\rho \left(\frac{2}{9}C_w T_s^2 L + \frac{4}{9}C_w \pi T_s^2 R\right) \tag{4.9}$$

The total cost for a single pressure vessel, without including equipment costs is

therefore,

$$C(L,R,T_s,T_h) = 2\pi\rho\left(C_s RT_s L + C_h R^2 T_h + \frac{2}{9}C_w T_s^2 L + \frac{4}{9}C_w \pi T_s^2 R + C_p T_s R(L_o - L)\right) \quad [4.10]$$

There is also a cost associated with ordering the material, C_{order}. Each time an order for raw material is placed, a fee of $250 is assessed in order to cover shipping, handling, and stocking in the inventory. This cost is based on the number of different sized sheets of raw material in order; it is not related to the quantity of sheets ordered. This cost is tabulated using Equation 4.11.

$$C_{order} = \sum_{i=1}^{m} \$250 \quad [4.11]$$

where m is the number of distinct sheets of metal required for the manufacturing process (i.e., the number of distinct values of L_o).

The cost C_t of manufacturing a different value of thickness amounts to an additional fee of $50 in order to make the necessary changes.

The cost of purchasing manufacturing equipment (C_{equip}), namely the forging presses and the associated dies to make the heads of the vessels, is evaluated with

$$C_{equip} = \sum_{p=1}^{N_p} \left(500000 + 50000R_p\right) + C_t \quad [4.12]$$

where N_p is the number of presses needed, and R_p is the radius of the die used with each press. The cost equation presented in Equation 4.3 is the total cost of producing one vessel; the ordering cost and equipment cost (Equations 4.11 and 4.12, respectively) are calculated for the entire family of vessels and their required manufacturing needs. It is assumed that each forging press has a very large production capacity; one press for each individual radius is sufficient to meet any demand.

This completes the modeling of cost. Now a manner for determining the selling price of each product variant is determined in order to find out the total profit. For this example the selling price is established as a function of volume and pressure:

$$Selling\ Price = C_v V + C_p P \qquad [4.13]$$

where C_v and C_p are the cost constants for volume (255 $/m^3) and pressure (255 $/MPa)

respectively. Finally, total profit is established by summing the difference of the selling price and cost for all of the product variants, n, as well as subtracting the cost of the orders, and the cost of the manufacturing equipment needed for the production process.

$$Profit_{total} = \sum_{i}^{n} \left[SellingPrice_i - C_i(L, R, T_s, T_h) \right] - C_{order} - C_{equip} \qquad [4.14]$$

4.1.3 Product Platform Constructal Theory Method for the Design of a Pressure Vessel Product Platform to deal with Uncertain Demand

This section helps to discuss the steps relevant to deal with uncertain demand in the PPCTM-RCM (augmented PPCTM) (Figure 3.11) by applying it to the pressure vessel problem. PPCTM-PLUS is useful in situations where the demand is well defined and there are no changes foreseen in future and so the platform does not give favorable results with uncertain demand. In this section, the pressure vessel platform is designed by using the PPCTM-RCM and the results are compared with PPCTM-PLUS to show that the platform is now robust to changes in demand.
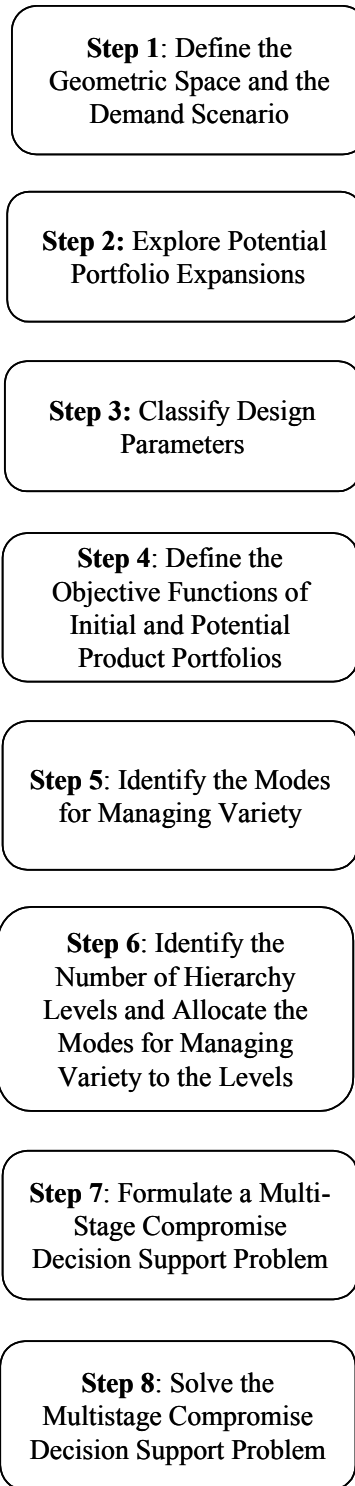
Figure 4.4: PPCTM-RCM (Augmented PPCTM)

Step 1: Define the Geometric Space and the Demand Scenario

Product customization is offered in two independent design specifications – the volume and pressure of the vessel. The resulting two-dimensional continuous space of customization along with the demand is shown in Figure 4.5.
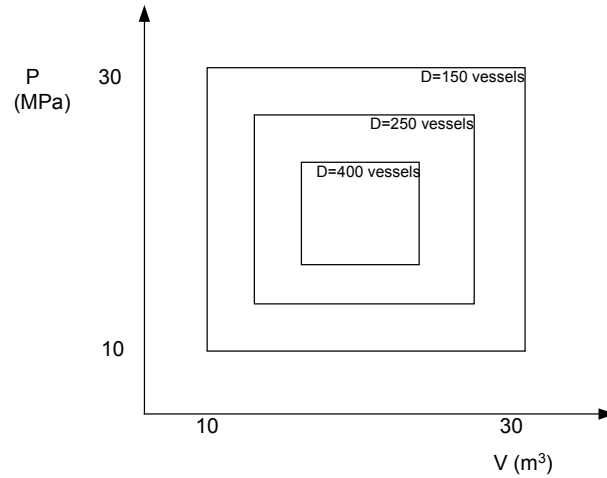
Figure 4.5: Pressure Vessel Space of Customization

The demand in the market space can take any form – discrete, concentric square, linear continuous, gaussian distribution or any random model of demand. It is assumed to be concentric square in this case.

Step 2: Explore potential portfolio expansions

In Section 4.1.1, it is assumed that in this chapter there are no changes in the extents of the marketplaces. Hence, there is no need of exploring potential portfolio expansions and so this step is skipped.

138

Step 3: Classify Design Parameters

This step begins the formulation of the problem by classifying the design factors. The classification is illustrated in Figure 4.6. Here, the demand is considered as the noise factor, while the control factors are the length, thickness of head and shell and radius of the vessel. Hence, providing some variation in demand to a product, in the form of noise, one can study the response i.e., profit obtained keeping the control factors constant.
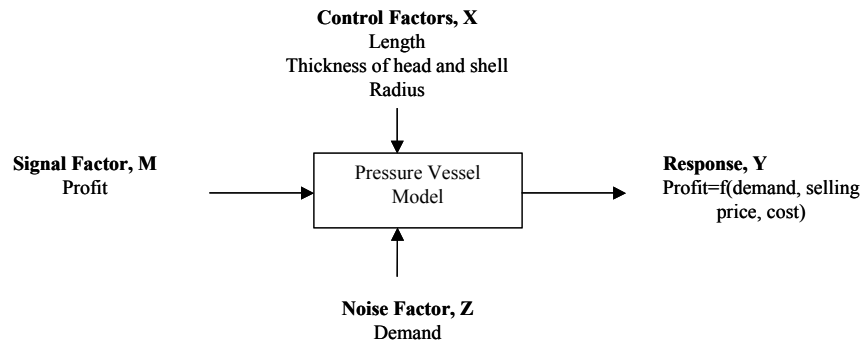
Figure 4.6: Factors and Response for Pressure Vessel Design

Step 4: Define the Objective Functions

The manufacturer has two conflicting goals – one is maximizing the profit and the other is to minimize the deviation in profit due to fluctuations in demand. The primary objective, maximizing the profit, is the goal that the manufacturer wishes to "move the mean to target."

The average profit *AvgProfit* is calculated by dividing the total profit obtained in Equation 4.15 by the total demand of the products of the market space.

$$AvgProfit = \frac{1}{D_{tot}} \left[\left(\sum_{i=1}^{n} D_i (SellingPrice_i - C_i) \right) - C_{order} - C_{equip} \right] \qquad [4.15]$$

where *SellingPrice$_i$* is the selling price of each product variant, C_i is the cost of each pressure vessel, C_{order} and C_{equip} are the costs associated with ordering and manufacturing equipment, respectively, and D_i is the demand of the specific product variant i, and D_{tot} is the total demand of the products of the market space.

As seen in Figure 4.6, demand is considered as the noise factor for this example. The other goal of this problem is to determine the control factors that minimize the variance of profit that typically arise from fluctuations in demand. The variance in profit is calculated by the following formula:

$$\sigma^2 = \sum (P_i - P)^2 / (n\text{-}1) \qquad [4.16]$$

where P_i is the profit calculated for demand D_i,

    P is the profit at mean demand, and

    n is the number of levels of the noise factor.

Considering the entire market space, the sum of variances of all considered products in the market space is calculated. The objective is to minimize the sum of variances for a particular product platform.

Step 5: Identify the Modes for Managing Variety

After brainstorming, designers have identified four different modes of managing product variety in pressure and volume.

Dimensional Customization of Shell Length (V1)

For this mode, the radius is kept constant while changes in volume are achieved by cutting each shell length from a stock piece of raw material. Hence, for a certain range of

volumes, there is a common stock length of material with the radius remaining same. While it is cheaper to offer the minimum required length for a given radius, there is an ordering cost associated with each different length of raw material.

Figure 4.7: Dimensional Customization of the Length of the Shell (Hernandez, 2001)

Commonalization of Radius (V2)

The changes in volume in this mode are achieved by modifying the radius of a vessel. Since each different radius used leads to a new press and a die purchase, there is an increase in equipment cost. At the same time, it is obvious that there also exists a cost savings in material and waste costs by having radii that closely match that which is required.

Modular Combination of Vessels (V3)

In this mode, modular combination of vessels is used to achieve discrete changes in volume. This is done by combining vessels of two lesser volumes. This mode affects the costs of manufacturing equipment as well as the material costs. There is a saving on the purchase of manufacturing equipment costs while the material costs associated with using two smaller vessels may offset this savings. Hence, in this mode one determines the range of volumes for which modularity is used.

141

Figure 4.8: Modular Combination of Vessels (Hernandez, 2001)

Standardization of Thickness of Head and Shell (P1)

In this mode of managing variety in pressure, the modifications in the thickness of both the shell and the head of the vessel provide a way of changing the pressure. It is based on the fact that if a shell with dimensions R and L, and thickness T_s and T_h satisfy the pressure constraints for a pressure P_1, it also satisfies these constraints for any pressure $P \leq P_1$. There is also an additional charge for use of different thicknesses.

Step 6: Identify the Number of Hierarchy Levels and Allocate the Modes for Managing Variety to the Levels

In this step, the different modes of managing product variety are ordered hierarchically.

The First Stage and the First Space Element

As stated in Section 2.1, the first stage in the hierarchy is held by the mode that offers the smallest variations in the design parameters. Hence, out of the different modes available for managing variety, the mode with the highest fidelity is selected for this stage. For this first space element, Modes V1 (dimensional customization of shell length)

and P1 (standardization of head and shell thickness) are used together to achieve variety in both volume and pressure. Dimensional customization of shell length makes small variations in volume possible. By cutting the raw material, of length L_o, to any length desired, L, this mode achieves continuous changes of volume.

The value of the variables ΔV_1 and ΔP_1 determines the size and shape of the first space element, S_1 (Figure 4.9). ΔV_1 and ΔP_1 represent the extent to which these modes of managing variety are applied to the platform. These need to be determined at this stage and are therefore the decision variables for this first stage.
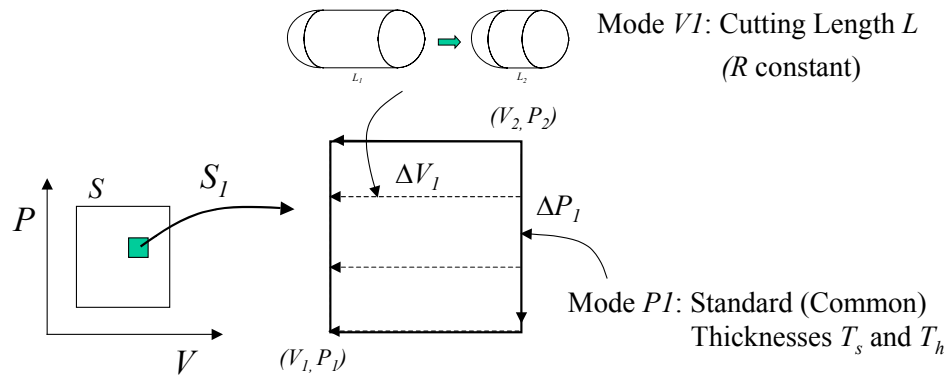
Figure 4.9: The First Space Element of the Pressure Vessel Space for Customization (Hernandez, 2001)

The Second Stage and the Second Space Element

The second space element, S_2, is composed by a number of first space elements, S_1, in the volume dimension. The mode after dimensional customization of shell length that offers small variation in volume is the commonalization of radius (Mode V2). Hence, for this second space element, commonalization of the radius (Mode V2) is used to achieve variety in volume. The value of the variable ΔV_2 determines the size and shape of the second space element, S_2 (Figure 4.10). ΔV_2 represents the extent to which the radius is

commonalized across the platform and is therefore decision variable for this second stage.
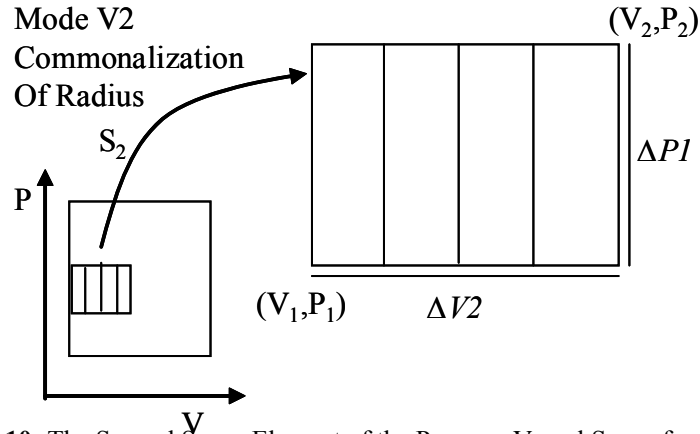
Figure 4.10: The Second Space Element of the Pressure Vessel Space for Customization

The Third Stage and the Third Space Element

The third space element, S_3, is composed by a number of second space elements, S_2, in the volume dimension. For this third space element, modularity of vessels (V3) is used to achieve variety in volume (Figure 4.11). The value of the variables ΔV_3 determines the size and shape of the third space element. ΔV_3 represents a point in the space in which modular combination of vessels begins, and is therefore decision variable for this third stage.

For example, if there exist customizable vessels in a range V_{min} to V', volumes larger than V' (up to $2V'$) are achieved by combining vessels within this range. In Figure 4.11, V' is 20 m^3, so all vessels with volumes greater than V' are achieved by combining two vessels. Hence, the variable to be determined in this case is the cut-off point (V') – a point where vessels with larger volumes are composed of two smaller vessels below the cut-off point.
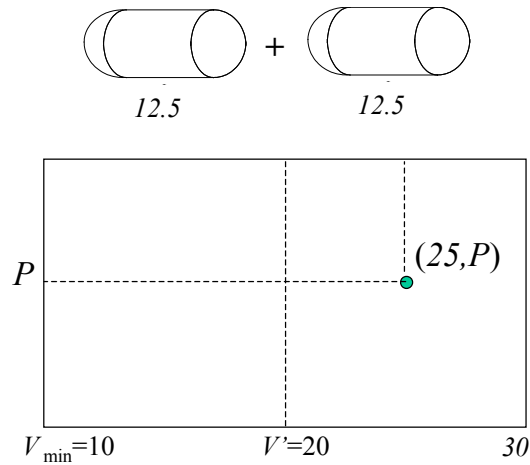
Figure 4.11: An Example of the Modular Combination of Vessels (Hernandez, 2001)

All the three subspaces with the decision variables are seen together in the Figure 4.12 below.
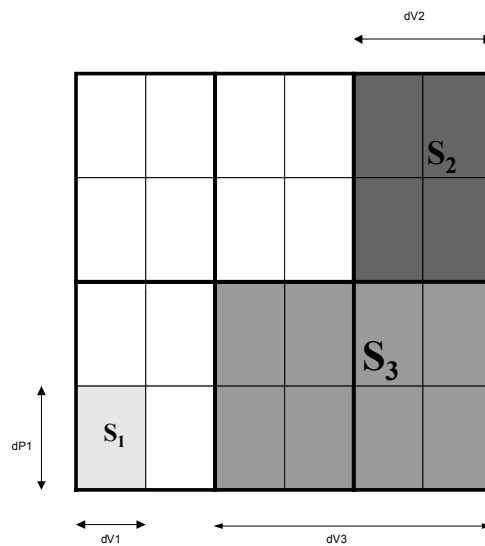
Figure 4.12: Product platform in terms of dV3, dV2, dP1, dV1

A graphical representation of the different modes of managing variety is seen in the Figure 4.13 below.
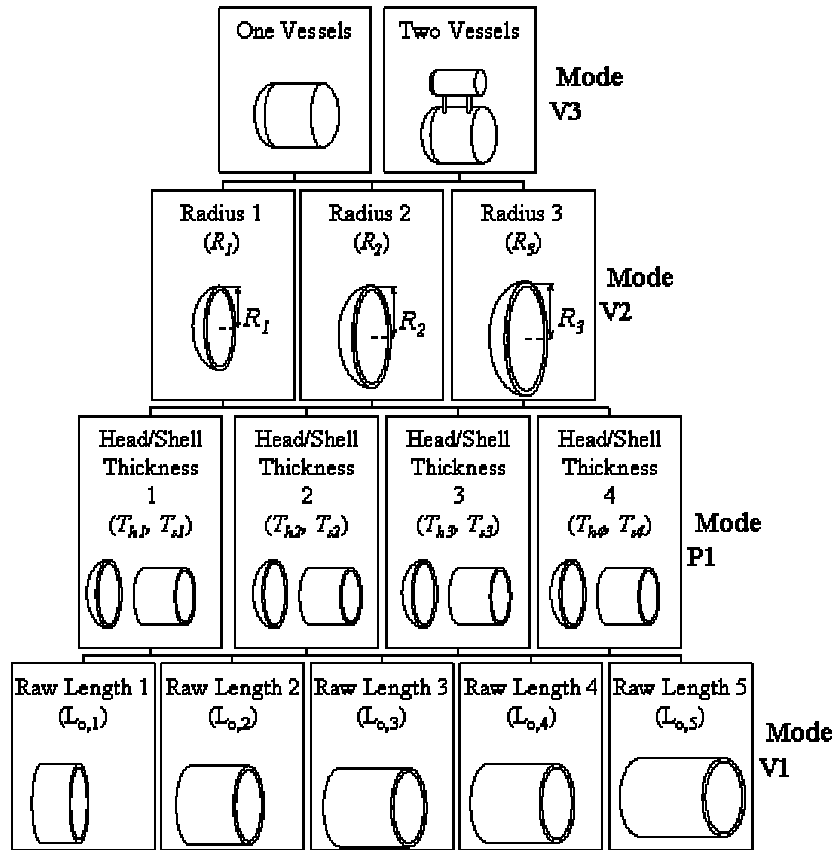
Figure 4.13: Hierarchic Organization of the Modes for Managing Product Customization for the Pressure Vessel (Williams, 2003)

Step 7: Formulate a Multi-Stage Compromise Decision Support Problem

There are two goals that need to be satisfied to get a robust design – maximizing profits and minimizing variation in profit. Since there are two goals, a decision has to be made that gives the best possible combination of both. The compromise DSP is used to formulate this problem. The system goals are formulated for each decision stage of PPCTM because their focus is in achieving these two goals. The goal for each stage is based on the individual objective functions of profit and variance. The values of profit and variance to be used are found using average value and summation respectively, calculated in each decision stage.

In the formulation of the compromise Decision Support Problem, the goals are measured in terms of the deviation of the objective function from the ideal value. In this case, the first objective is of maximizing profit and the second objective is to minimize variation. As discussed in Section 3.1, the average profit *AvgProf* is divided by the maximum possible profit *IP* and the deviation of this ratio from the ideal value of 1 is calculated. Similarly, the minimum possible variance *IV* is divided by the actual variance and the deviation of this ratio from the ideal value of 1 is calculated. Each set of objective function is represented by:

$$(AvgProf / IP) + d_i^- - d_i^+ = 1 \qquad\qquad [4.10]$$

$$(IV / Variance) + d_i^- - d_i^+ = 1 \qquad\qquad [4.11]$$

The deviation variables are thus found. Using the weights of each objective as discussed above, along with the deviation variables forms the deviation function.

$$Z = k_{AvgProfit}\, d_1^- + k_{variance}\, d_2^-$$

The First Stage and the First Space Element

The decision variables of the first space element are ΔV_1 and ΔP_1, the range of application of Modes V1 and P1 respectively. Three design variables are commonalized in the first space element through the application of these two modes of managing customization: the raw length, L_o, the head thickness, T_h, and the shell thickness, T_s.

<u>Decision 1: ΔV_1, ΔP_1</u>

Given: The two-dimensional market space S = {(V, P)}

Mode V1: Dimensional Customization of the Shell Length

Mode P1: Standardization of the Head and Shell Thickness

Mean and variance of the noise factor

Find: The value of decision variable ΔV_1, ΔP_1

The deviation variables, d_1^- and d_1^+, d_2^- and d_2^+

Satisfy: Bounds: $0 \le \Delta V_1 \le 20$

$0 \le \Delta P_1 \le 20$

Constraints: $T_s \ge \left(\dfrac{P}{\sigma_y - 0.6P} \right) R$, $\quad T_h \ge \left(\dfrac{P}{2\sigma_y - 0.2P} \right) R$

$d_1^-, d_1^+ \ge 0 \quad ; \quad d_1^- . d_1^+ = 0$

$d_2^-, d_2^+ \ge 0 \quad ; \quad d_2^- . d_2^+ = 0$

Goals: $(AvgProf / IP) + d_1^- - d_1^+ = 1$

$(IV / Variance)] + d_2^- - d_2^+ = 1$

Minimize: $Z = k_{AvgProfit}\, d_1^- + k_{variance}\, d_2^-$

where: *AvgProf* is calculated as in [4.15] for market space

IP = Maximum profit of market space

Variance = Variance of market space

IV = Minimum variance of market space

Figure 4.14: Formulation of the Multi-Stage cDSP for the First Space Element

The Second Stage and the Second Space Element

The decision variable of the second space element is ΔV_2, range of application of Mode V2. Only design variable, radius R is commonalized in the second space element.

<div style="border:1px solid black; padding:10px;">

<center><u>Decision 2: ΔV_2</u></center>

Given: The two-dimensional market space S = {(V, P)}

The value of ΔV_1, and ΔP_1 from the first stage

Mode V2: Commonalization of the Radius

Mean and variance of the noise factor

Find: Value of decision variable ΔV_2 , deviation variables, d_3^- d_3^+, d_4^- , d_4^+

Satisfy: Bounds: $0 \leq \Delta V_2 \leq 20$,   Constraints: $\Delta V_1 \leq \Delta V_2 \leq 20$

$d_3^-, d_3^+ \geq 0 \,;\, d_3^- \,.\, d_3^+ = 0\,;\, d_4^-, d_4^+ \geq 0 \,;\, d_4^- \,.\, d_4^+ = 0$

Goals: $(AvgProf / IP) + d_3^- - d_3^+ = 1$

$(IV / Variance) + d_4^- - d_4^+ = 1$

Minimize: $Z = k_{AvgProfit}\, d_3^- + k_{variance}\, d_4^-$

where: *AvgProf* is calculated as in [4.15] for market space

IP = Maximum profit of market space

Variance = Variance of market space

IV = Minimum variance of market space

</div>

<center>**Figure 4.15:** Formulation of the Multi-Stage cDSP for the Second Space Element</center>

The Third Stage and the Third Space Element

The third and final space element is defined by the range of application of Mode V3: Modular Combination of Vessels. The decision associated with this mode is the selection of "cut-off" point – a point where vessels with larger volumes are composed of two smaller vessels below the cut-off point. The space element S_3 is composed of a number of spaces S_2 from the minimum volume of the space, V_{min}, to a cut-off point, V'.

Decision 3: ΔV_3

Given: The two-dimensional market space S = {(V, P)}

             The value of ΔV_1, ΔP_1, ΔV_2 from the first and second stages

             Mode V3: Modular Combination of Vessels

             Mean and variance of the noise factor

Find: Value of decision variable ΔV_3 , deviation variables, d_5^- , d_5^+, d_6^- , d_6^+

Satisfy: Bounds: $0 \leq \Delta V_3 \leq 20$, Constraints: $\Delta V_1 \leq \Delta V_2 \leq \Delta V_3 \leq \leq 20$

                      $d_5^-, d_5^+ \geq 0$; $d_5^- . d_5^+ = 0$; $d_6^-, d_6^+ \geq 0$; $d_6^- . d_6^+ = 0$

           Goals: $(AvgProf / IP) + d_5^- - d_5^+ = 1$

                       $(IV / Variance) + d_6^- - d_6^+ = 1$

Minimize: $Z = k_{AvgProfit} \, d_5^- + k_{variance} \, d_6^-$

              where: *AvgProf* is calculated as in [4.15] for market space

                       IP = Maximum profit of market space

                       Variance = Variance of market space

                       IV = Minimum variance of market space

Figure 4.16: Formulation of the Multi-Stage cDSP for the Third Space Element

With the multi-stage cDSP formulated, we move on to the solution of the augmented PPCTM, which is the last step.

Step 8: Solve the Multi-Stage Compromise Decision Support Problem

There are two ways in which this problem can be tackled – continuous or discrete analysis. In continuous analysis, one needs to express the objectives in terms of the design parameters. This includes expressing the demand in terms of the design parameters. As identified in (Williams, 2003), the continuous analysis becomes complex and mathematically demanding due to the consideration of multiple objectives and changing demand.

In discrete analysis, the analysis is done on a discrete set of points in the space which helps to approximate the entire space. For discretization of the space, a certain resolution is chosen by the designer. Considering this resolution, nodes are established in the space of customization and objective function at every node is calculated. This method was also used in (Williams, 2003) and seems to be appropriate for such problems in which expressing the objective function in terms of the design variables is difficult due to consideration of multiple objectives and changing demand.

Now the formulation of an appropriate solution technique needs to be determined. Any appropriate solution algorithm can be used; its primary goal is the determination of the extent of application of the modes that provides the least deviation from the ideal value, from which the commonality of design parameters of the product platform is determined.

Our solution technique is illustrated in Figure 4.17 (modified from Williams, 2003). This solution method involves iterating through values of the modes of managing variety

(ΔV_1, ΔV_2, ΔV_3, and ΔP_1), establishing the dimensions of the sub-spaces, commonalizing the design parameters (L_o, T_s, T_h, and R) across each sub-space, evaluating the objective functions, and calculating and comparing the deviation function of each iteration. Specifically:

-    A particular combination of sub-space element sizes is taken as input. The input also includes demand scenario, geometric space (market space) and the objective functions (maximize profit, minimize variance).

-    From the sizes of the sub-space elements, the bounds of each sub-space are established. The values of parameters to be commonalized are determined from the bounds of the sub-space. This involves calculation of the parameters from their respective modes.

-    These parameters are then commonalized across the respective sub-spaces. The objective function is determined for every variant considering the parameter values across each sub space.

-    For calculation of the variance in profit, the first step involves calculation of cost from the parameter values and assessing the selling price and demand for each variant. Then variance is provided to demand in the form of noise. The profit is then calculated while keeping other parameters constant. From the profit obtained from different values of demand, the variance in profit is calculated. It is assumed in this problem that the demand can vary ±20% from the predicted snapshot of demand. Hence, ±20% noise is provided to the values of demand in four discrete levels (i.e., Demand (Noise) = {0.8D, 0.9D, 1.1D, 1.2D}, where D is the mean demand), this is illustrated in Figure 4.18.

-   After obtaining the variance in profit for each variant, it is added up to get the total variance. Similarly, the profit of each variant is summed and averaged across the space.

-   The values of variance and profit help to obtain the deviation function as the output of this algorithm.

-   The value of the deviation function is returned from the analysis and is compared with previous output. After each feasible combination of ranges of the modes for managing variety is explored, the input that provides the smallest deviation from the goal is selected.
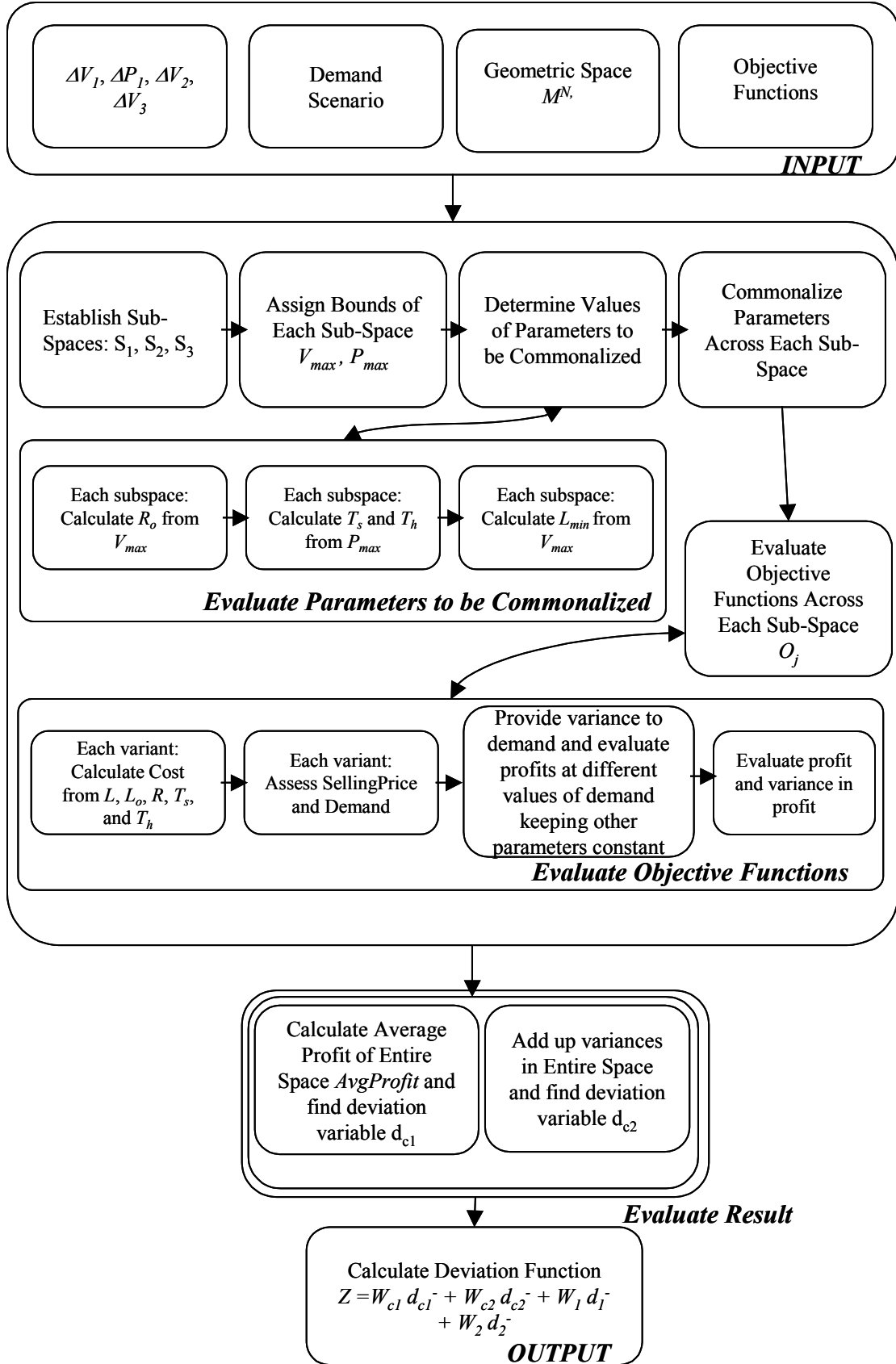
Figure 4.17: Solution Algorithm for Minimizing Variance in the Pressure Vessel example

154
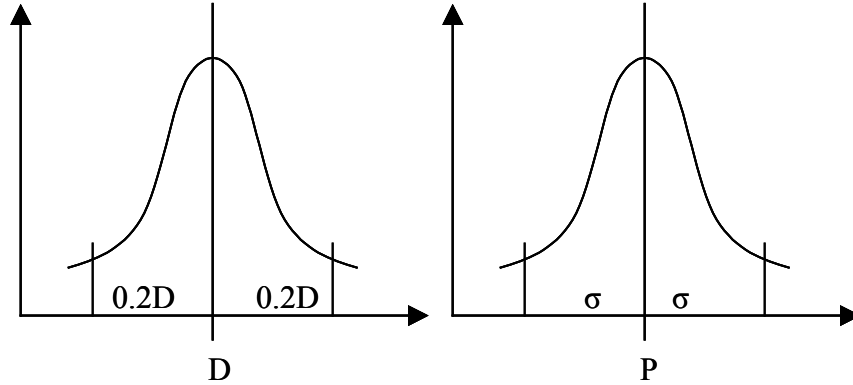
Figure 4.18: Demand and Profit – Mean and Standard Deviation

As the pressure vessel example is relatively simple, and reasons discussed in Section 3.3.8, an exhaustive search of different combinations of values of ΔV_1, ΔV_2, ΔV_3, and ΔP_1 is done. It is illustrated in Figure 4.19. The input for the solution algorithm is different combinations of the ranges of the modes. After taking the ranges of the modes, the solution algorithm explained above calculates the value of the deviation function. This deviation function value is compared with the previous values. This is continued for different combinations of the ranges of the modes. The instantiation of this algorithm for the pressure vessel example problem is provided in Appendix A as a Java computer program.

The main benefit of using the exhaustive search is that one is able to explore all the possible combinations of the solutions. The number of computations is manageable and hence benefit of exploration of the entire space outweighs the computation intensity of the method. Moreover, this has been observed in (Williams, 2003) by comparing with techniques other than exhaustive search.
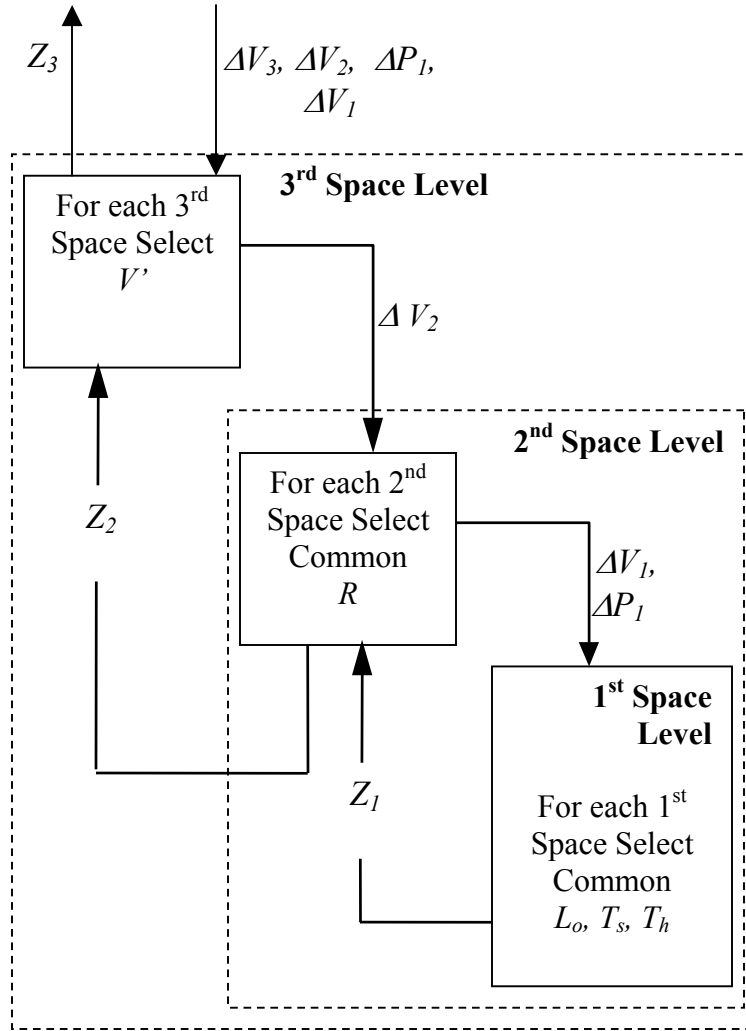
Figure 4.19: Exhaustive Search of Design Parameters for the Pressure Vessel Example (Modified from Williams, 2003)

The formulation of the PPCTM for the pressure vessel example is completed and its solution algorithm is outlined. The Section 4.1.4 presents the results.

4.1.4   Results – Scenarios in the Pressure Vessel Problem

The objective of using this example problem is to test the section of the augmented method that handles uncertain demand. So in order to test it, the demand is varied from values in between 0.8 and 1.2 of its mean value. Given this variance in demand, the

product platform is designed using the above-discussed augmented method. Results in terms of the mean profit and the variance in profit of the product platform due to changing demand are observed.

In order to better observe the effects of the infusion of robust design principles, there are three scenarios in which the pressure vessel product platform is designed. In the first scenario, there is non-uniform demand; however there is no uncertainty in the distribution of demand. In the second scenario, there is not only non-uniform demand but also there is a certain uncertainty in the distribution of demand. The third scenario is similar to the second scenario in terms of demand; however, the weights assigned to the objectives of the product platform are changed and results analyzed for comparison purposes.

Scenario 1 – Certain Distribution of Demand

In this scenario, it is assumed that there is a predicted snapshot of demand and there are no changes expected in this distribution of demand. In other words, there is non-uniform demand; however there is no uncertainty in the distribution of demand. This is a similar situation as in (Williams, 2003). In this case, the weight assigned to the objective of 'minimizing the variance' is given zero, while the weight given to other objective of maximizing profits, i.e., 'bringing the mean to target' is one. This gives similar results as obtained by using PPCTM-PLUS in (Williams, 2003).

Table 4.1: Scenario 1 Product Platform Results

Scenario #	$k_{variance}$	k_{profit}	$\Delta V3$ (m^3)	$\Delta V2$ (m^3)	$\Delta P1$ (MPa)	$\Delta V1$ (m^3)	Profit ($)	Total Variance (^{2})	Z
1	0	1	20	5	0.5	5	4698.75	4.22E+16	0.00027

The large values of variances in each of the scenarios are due to three factors. Firstly, since variance in profit for every product is considered, the quantity gets magnified by the demand of every product. Secondly variance as stated in the formula is the square of the difference in varied profit and profit from mean demand. Thirdly, the total variance is the sum of the variances of every product in the product platform.
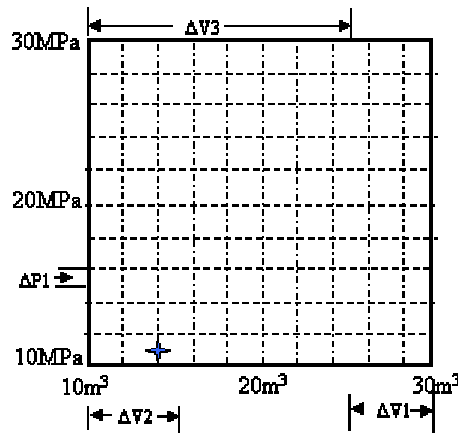
Figure 4.20: Platform ΔV1, ΔP1, ΔV2 and ΔV3 – Scenario 1 Product Platform Results
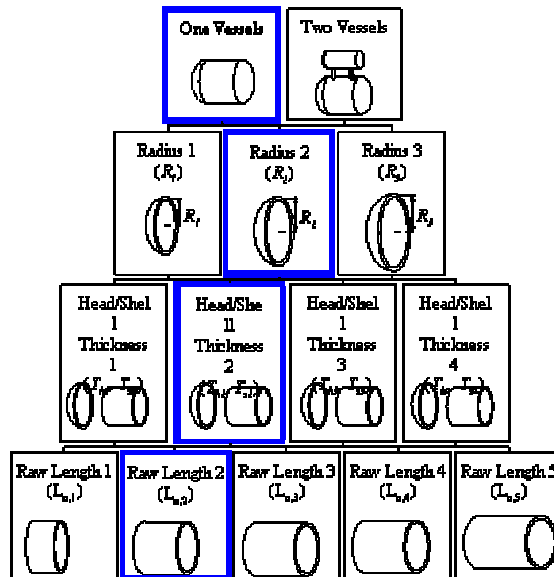
Figure 4.21: Hierarchical Arrangement of the Product Platform

Table 4.2: Section of the Scenario 1 Product Platform Specifications

V	P	R (m) ($\Delta V2=5m^3$)	Ts (mm) ($\Delta P1=0.5MPa$)	Th (mm) ($\Delta P1=0.5MPa$)	Lo (m) ($\Delta V1=5m^3$)
	10		7.311	3.638	
	10.5		7.679	3.821	
	11		8.047	4.003	
	11.5		8.415	4.185	
	12		8.783	4.367	
	12.5		9.151	4.549	
	13		9.520	4.731	
	13.5		9.889	4.914	
	14		10.258	5.096	
	14.5		10.628	5.278	
	15		10.997	5.460	
	15.5		11.367	3.638	
	16		11.737	3.821	
	16.5		12.107	4.003	
	17		12.478	4.185	
	17.5		12.848	4.367	
	18		13.219	4.549	
	18.5		13.590	4.731	
	19		13.961	4.914	
	19.5		14.333	5.096	
15	20	0.783	14.704	7.284	6.744
	20.5		15.076	5.460	
	21		15.448	5.643	
	21.5		15.820	5.825	
	22		16.193	6.007	
	22.5		16.566	6.189	
	23		16.938	6.372	
	23.5		17.312	6.554	
	24		17.685	6.737	
	24.5		18.058	6.919	
	25		18.432	9.109	
	25.5		18.806	7.284	
	26		19.180	7.466	
	26.5		19.555	7.649	
	27		19.929	7.831	
	27.5		20.304	8.014	
	28		20.679	8.196	
	28.5		21.054	8.379	
	29		21.430	10.570	
	29.5		21.805	8.744	
	30		22.181	10.936	

In Table 4.1, the values of ΔV_1, ΔV_2, ΔV_3, and ΔP_1 in Scenario 1 define a product family; the corresponding profit, variance and deviation Z of that particular family are

159

noted in the last three columns. To assist the reader with the interpretation of ΔV_1, ΔV_2, ΔV_3, and ΔP_1 as a product family, the following tutorial is provided. The results inform the manufacturer that the best configuration of the modes of managing variety which closely matches his/her preferences is to commonalize the raw length for every 5 m^3 of volume, commonalize the head and shell thickness for every 0.5 MPa of pressure, commonalize the radius for every 5 m^3, and to not modularly combine the vessels. It is seen that the market space is divided into subspaces and the size of the subspaces depend on the values of ΔV_1, ΔV_2, ΔV_3, and ΔP_1 (Figure 4.20).

Consider a customer having volume and pressure requirements of a pressure vessel as 14m^3 and 11MPa respectively (denoted by star in Figure 4.20). Using the different modes of managing variety at each hierarchical level (Figure 4.21), one obtains the product platform specifications involving R, T_s, T_h and L from the particular values of ΔV_1, ΔV_2, ΔV_3, and ΔP_1. A section of the design specification of the product family of this scenario is presented in Table 4.2. The columns in Table 4.2 represent the extent of commonalization for each design variable. It is observed that the figures in bold in Table 4.2 satisfy the requirements of the customer. The detailed specifications of the product platform of this scenario are provided in Appendix B.

Scenario 2 – Uncertain Distribution of Demand - A

In this scenario, there is not only non-uniform demand but also there is a certain uncertainty in the distribution of demand. The demand distribution is varied in between 0.8 and 1.2 of its mean value. Given this variance in demand, the product platform is designed. The method used is the PPCTM-RCM as discussed in Section 3.3. The weight assigned to both the objectives is equal. The results for this scenario are seen in Table

4.3. Two customer examples are provided (Table 4.4 and Table 4.5). The specific values of the design variables are provided as results of the application of the PPCTM to this example.

Table 4.3: Scenario 2 Product Platform Results

Scenario #	$k_{variance}$	k_{profit}	ΔV3 (m³)	ΔV2 (m³)	ΔP1 (MPa)	ΔV1 (m³)	Profit ($)	Total Variance ($²)	Z
2	0.5	0.5	20	2.5	2	0.5	4680.51	4.14E+16	0.38110
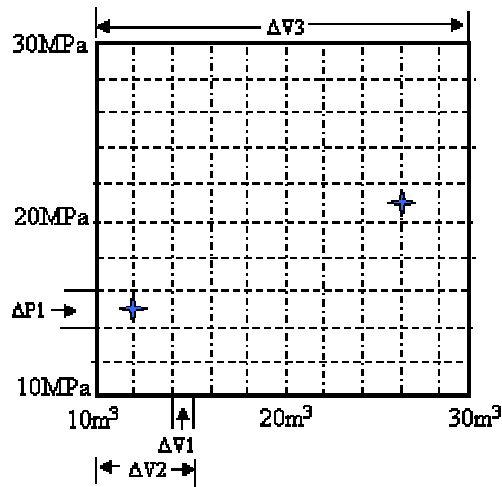
Figure 4.22: ΔV1, ΔP1, ΔV2 and ΔV3 – Scenario 2 Product Platform Results

Table 4.4: Section of the Scenario 2 Product Platform Specifications (12m³ and 15MPa)

V	P	R (m) (ΔV2=2.5m³)	Ts (mm) (ΔP1=2MPa)	Th (mm) (ΔP1=2MPa)	Lo (m) (ΔV1=0.5m³)
12	10	0.718	6.701	3.335	6.459
	12		8.050	4.003	
	14		9.403	4.671	
	16		10.758	5.339	
	18		12.116	6.008	
	20		13.478	6.676	
	22		14.842	7.345	
	24		16.210	8.015	
	26		17.581	8.684	
	28		18.955	9.354	
	30		20.331	10.024	

161

Figure 4.23: Hierarchical Arrangement of the Product Platform (12m^3 and 15MPa)

Table 4.5: Section of the Scenario 2 Product Platform Specifications (26m^3 and 21MPa)

V	P	R (m) (ΔV2=2.5m^3)	Ts (mm) (ΔP1=2MPa)	Th (mm) (ΔP1=2MPa)	Lo (m) (ΔV1=0.5m^3)
26	10	1.043	9.739	4.847	6.217
	12		11.699	5.817	
	14		13.665	6.788	
	16		15.634	7.759	
	18		17.608	8.730	
	20		19.587	9.702	
	22		21.570	10.675	
	24		23.557	11.647	
	26		25.549	12.620	
	28		27.546	13.593	
	30		29.547	14.567	

Figure 4.24: Hierarchical Arrangement of the Product Platform (26m^3 and 21MPa)

The results help the manufacturer in the following manner. The best configuration of the modes of managing variety which closely matches his/her preferences is to commonalize the raw length for every 2.5 m^3 of volume, commonalize the head and shell thickness for every 2 MPa of pressure, commonalize the radius for every 0.5 m^3, and not to modularly combine the vessels (Table 4.3).

Here, two customer examples are shown.

- Consider a customer having volume and pressure requirements of a pressure vessel as 12m^3 and 15MPa respectively (denoted by left star in Figure 4.22). Using the different modes of managing variety at each hierarchical level (Figure 4.23), one obtains the product platform specifications involving R, T$_s$, T$_h$ and L from the particular values of ΔV_1, ΔV_2, ΔV_3, and ΔP_1. A section of the design specification of the product family of this scenario is presented in Table 4.4. It is observed that the figures in bold in Table 4.4 satisfy the requirements of the

163

customer. The vessel of the product family that will satisfy the customer requirements is the one having volume as $12m^3$ and pressure as 16MPa. The radius of this vessel is 0.718m, thickness of shell is 10.758mm, thickness of head is 5.339mm and raw length of sheet is 6.459m.

- Consider another customer having volume and pressure requirements of a pressure vessel as $26m^3$ and 21MPa respectively (denoted by right star in Figure 4.22). A section of the design specification of the product family of this scenario is presented in Table 4.5. It is observed that the figures in bold in Table 4.5 satisfy the requirements of the customer. The vessel of the product family that will satisfy the customer requirements is the one having volume as $26m^3$ and pressure as 22MPa. The radius of this vessel is 1.043m, thickness of shell is 21.57mm, thickness of head is 10.675mm and raw length of sheet is 6.217m.

It is important to note in both the examples the customer is satisfied by having the pressure and volume of a pressure vessel higher than his/her requirements. Table in Appendix B serves as a "roadmap" for a designer; any potential combination of pressure and volume for the considered space of customization is connected to the specific design parameters, including dimensions of raw material. Both the above mentioned examples (Table 4.4 and Table 4.5) are extracted from table in Appendix B and are highlighted in bold.

Scenario 3 – Uncertain Distribution of Demand - B

The third scenario is similar to the second scenario in terms of demand; however, the weights assigned to the objectives of the product platform are changed and results analyzed for comparison purposes. The weight assigned to the costs objective is zero, while the weight assigned to the variance objective is one.

Table 4.6: Scenario 3 Product Platform Results

Scenario #	$k_{variance}$	k_{profit}	$\Delta V3$ (m^3)	$\Delta V2$ (m^3)	$\Delta P1$ (MPa)	$\Delta V1$ (m^3)	Profit ($)	Total Variance $($^2)$	Z
3	1	0	10	10	10	10	1297.59	1.35E+16	0.25780

4.1.5   Analysis of the Results

A comparative study similar to the one done in (Mistree, et al., 2002) has been presented. The design that is most unaffected by changes in demand, i.e., the most robust design is the one with the least variances in Table 4.1, Table 4.3 and Table 4.7. Increasing robustness is measured by the magnitude of the variance of profit, and the degree of robustness in a design is adjusted by varying the weights represented by $k_{AvgProfit}$ and $k_{variance}$ on both mean and variance. It is observed that in the first scenario (Table 4.1) the weight given to the variance objective is zero and hence it is the least robust design having variance as 4.22E+16. The first scenario design however offers higher profits $4,698.75.   As the weight of the variance objective increases, the robustness increases to have maximum robustness in the third scenario with variance as

165

1.35E+16 (Table 4.7). In the third scenario design, weight placed on the goal of maximizing profit is zero; thus, this design is generated considering only minimizing variance. Hence, the third scenario design tends to be more robust to variation in the demand; however, it exhibits lower average profits ($1,297.59). In Scenario 2, a natural trade-off between the mean and variance of profit is obtained as seen in Table 4.3 with profit as $4,680.51 and variance as 4.14E+16. Thus, it is seen that according to the manufacturer's preference for "moving the mean to target" or "minimizing deviation", the weights can be changed and the augmented PPCTM would give appropriate results.

The value of $\Delta P1$ is dependent on the mode of standardizing thickness of head and shell. There is a tradeoff between less cost due to the use of same thickness material and more cost due to use of more material. However, the cost saved by same thickness material is much less as compared to cost saved by using less material. Hence, when cost is given maximum weight in Scenario 1, the lowest possible range of commonalization for thickness is selected. As the weight of cost is reduced in Scenario 2 and variance is also considered, the range of commonality of $\Delta P1$ increases.

For Scenarios 1 and 2, the value of $\Delta V3$ remains constant at 20 and hence commonalization across the entire space is suggested. The reason is that the equipment costs are very high to enable the use of more than one vessel to satisfy the requirement. However, in Scenario 3, when profit is not at all considered due to zero weight, the value of $\Delta V3$ changes to achieve the variance objective.

Apart from the above discussed scenarios, the sensitivity of the results to different weights is studied. A scenario is run in which the weight is in between 0 and 0.5 for maximizing profit and in between 0.5 and 1 for minimizing variance. This scenario is

seen in Table 4.7 where the weight for maximizing profits is 0.35 and weight for minimizing variance is 0.65. It is observed that the profit obtained ($2859.49) is less than the profit obtained ($4680.51) in Scenario 2 and variance (1.77E+16) is less than the variance obtained (4.14E+16) in Scenario 2. Also, the profit obtained ($2859.49) is more than the profit obtained ($1297.59) in Scenario 3 and variance (1.77E+16) is more than the variance obtained (1.35E+16) in Scenario 3. The reason is that the weight in this scenario is in between that of Scenario 2 and Scenario 3.

Table 4.7: Scenario 4 Product Platform Results

Scenario #	$k_{variance}$	k_{profit}	ΔV3 (m³)	ΔV2 (m³)	ΔP1 (MPa)	ΔV1 (m³)	Profit ($)	Total Variance ($²)	Z
4	0.65	0.35	20	10	10	10	2859.49	1.77E+16	0.41940

Another scenario is run in which the weight is in between 0.5 and 1 for maximizing profit and in between 0 and 0.5 for minimizing variance. In this case the product platform obtained is the same as in Scenario 2 for values of weights close to that scenario while the product platform obtained is the same as in Scenario 1 for values of weights close to that scenario. The values of profit and variance in Scenario 1 and 2 are close to each other and hence change of weight does not lead to development of a new product platform. This can be better analyzed by considering the variation of the two objectives for different weights.
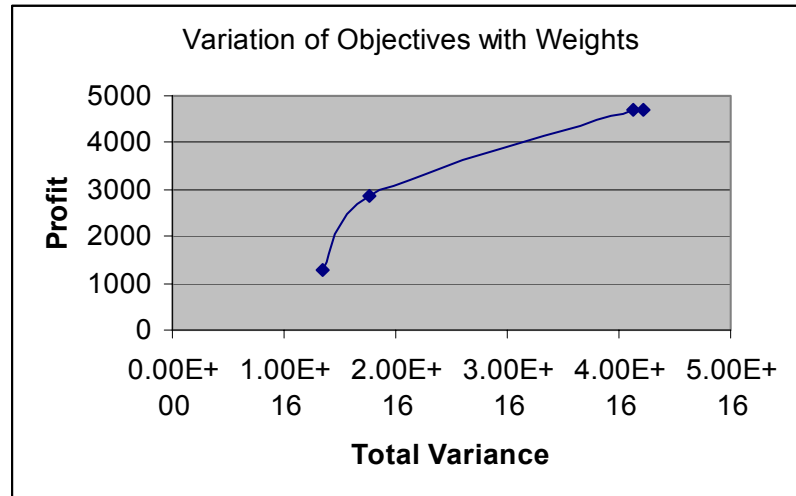
Figure 4.25: Plot of Objectives for Different Settings of Weights

The two objectives of maximizing profit and minimizing variance are plotted for different settings of the weight (Figure 4.25). It is observed that the slope of the line is greater in the range of weight 0 to 0.5 for maximizing profits while it is horizontal for 0.5 to 1 for same objective. This was observed in Scenario 1, 2, 3 and 4. There were more changes in the objectives in the range of 0.5 to 1 for maximizing profits than 0 to 0.5. Based on this plot and his/her preferences, the manufacturer can choose the appropriate weights to design the product platform.

Having seen the results for different scenarios, let us now find out what is the economic gain by designing according to the augmented method. Two more scenarios are analyzed to highlight the profits that a company makes by using this method.

4.1.6 Economic Gains using the method

Scenario A – Product platform without considering uncertain demand

Table 4.8: Profit Comparison without Augmentations

Scenario #	$k_{variance}$	k_{profit}	ΔV3 (m³)	ΔV2 (m³)	ΔP1 (MPa)	ΔV1 (m³)	Profit ($)	Total Variance ($²)	Z
A.1	0	1	20	5	0.5	5	4698.75	4.22E+16	0.00027
A.2							1875.28		
						Difference	2823.5		

Consider a manufacturer of pressure vessels who wishes to maximize his/her profits. He assumes that the demand is constant and thus has only one objective in mind to design a platform that will maximize his profits. The result is the Scenario A.1, which is the same as Scenario 1 in Table 4.1.

Now, consider that the demand of pressure vessels has reduced to 40%. Now the profit obtained by the manufacturer obviously reduces as shown in Scenario A.2; however, the point to note is the amount of difference between the two profits. This is analyzed in more detail while considering the difference in profits of Scenario B.

Scenario B – Product platform considering uncertain demand

Table 4.9: Profit Comparison with Augmentations

Scenario #	$k_{variance}$	k_{profit}	ΔV3 (m³)	ΔV2 (m³)	ΔP1 (MPa)	ΔV1 (m³)	Profit ($)	Total Variance ($²)	Z
B.1	0.5	0.5	20	2.5	2	0.5	4680.51	4.14E+16	0.381
B.2							1863.79		
						Difference	2816.7		

Now consider a manufacturer of pressure vessels who is aware of the uncertain demand and hence designs the product platform accordingly. The design of the product

platform takes place by considering both the objectives of increasing the profit and reducing the variance in costs due to changing parameters. The resulting product platform is seen in Scenario B.1 along with the costs associated with it. This is the same scenario as in Scenario 2 of Table 4.3.

Consider that the demand has changed to similar conditions as in Scenario A.2. Now if the manufacturer uses the existing product platform to incorporate these changes the profits associated with it are noted down in Scenario B.2. The profits are lower; however the point to note is the amount of difference between the two profits.

Now one sees the difference between Scenario A and B. The difference in costs of Scenario B is less than that of difference in costs of Scenario A. Hence it is observed that if the product platform is designed considering the uncertain demand, the costs associated with using the existing product platform for the changed demand later on will be less. Thus the product platform in Scenario B is robust to such changes in demand and will give favorable results despite of unforeseen changes in market.

The validity of results is established by three ways:

1. Checking the constraints
2. Comparison of a robust and non robust product platform
3. Minimizing variance in objectives due to uncertain demand leads to more commonality

1. **Checking the constraints -** There are three constraints in the problem. Firstly, the range of modes at lower levels must be smaller than the range of modes at higher

170

levels. This is seen from Table 4.10 that all the lower levels (e.g., V1) are smaller than the higher levels (e.g., V2). Secondly, the constraint that $d_i^-, d_i^+ \geq 0$ needs to be satisfied. Again, it is seen all the values in the Table 4.10 show that $d_i^-, d_i^+ \geq 0$. The third constraint is that $d_i^- \cdot d_i^+ = 0$. From the column of d+, it is seen that all the values are zero. Thus, one always obtains the product of d+ and d- as zero.

Table 4.10: Constraint Verification

ΔV3 (m³)	ΔV2 (m³)	ΔP1 (MPa)	ΔV1 (m³)	Profit		Variance	
				d-	d+	d-	d+
20	5	2	0.5	0.00624	0	0.7569	0
20	10	1	10	0.07194	0	0.73172	0
20	1	2	0.5	0.00647	0	0.7588	0
10	1	2	0.5	0.2782	0	0.6382	0
20	1	2	1	0.01148	0	0.7568	0
10	2	2	1	0.2838	0	0.6326	0
20	4	2.5	0.5	0.0207	0	0.7504	0
10	10	5	10	0.5057	0	0.4418	0

2. **Comparison of a robust and non-robust product platform** – The scenarios A and B help to validate the results. In these scenarios, one actually validates the results in Scenarios 1, 2 and 3. Scenario 1, 2 and 3 show that the reduction in variance in profit leads to less effect on profit due to changes in demand. The scenario A shows a non-robust product platform and the effect of having an uncertain demand on it is noted. Now the platform is made robust in Scenario B. Again the effect on the platform due to the change in demand is determined. According to the model as shown in Scenario 1, 2 and 3, since the reduction in variance in profit leads to less effect on profit due to changes in the demand, one should obtain results such that the robust product platform in Scenario 2 should

have lesser change in cost than the non-robust product platform in Scenario 3. This is exactly what the results indicate.

3. **Minimizing variance in objectives due to uncertain demand leads to less variety -** It is seen in the pressure vessel example for Scenario 3 in Table 4.7 there are large values of each of the decision variables, ΔV_1, ΔP_1, ΔV_3, ΔV_2. This implies that there is more commonality in this scenario where the only objective is to minimize the variation (since the other objective of bringing the mean to target has a weight of 0). One can try to analyze these results in more detail. For instance, consider the case that the manufacturer goes for less commonality and more variety in such an uncertain demand. Now, it has been shown in (Williams, 2003) and logically one can say that large demand leads to more variety. However, if this demand is highly uncertain, more variety will lead to unnecessary losses for the manufacturer due to the possible decrease of demand in future. So the manufacturer would not opt for taking the risk of providing a large product line but stick to fewer products. Hence the results make sense.

In this example problem, demand is varied and the platform is designed in such a situation using the augmented method. Appropriate results are obtained depending on the weights assigned to each goal. Also, the platform is designed without using this method and with using this method and it is observed that better economic gains are obtained in the latter case. Thus, after this example problem, the section of the method that deals with

uncertain demand has been tested. Some more insight and validity with respect to this example is provided towards the end of this chapter

In the next section, the method is applied to a line of customizable hand exercisers. The method is now tested if it can handle changing design parameters.

4.2 ROBUST DESIGN OF A FAMILY OF CUSTOMIZABLE HAND EXERCISERS TO DEAL WITH CHANGING DESIGN PARAMETERS (TYPE II ROBUST DESIGN PROBLEM)

In this section, the Q1.2 is addressed. The method is tested if it gives robust results to changing design parameters. So now it becomes a Type 2 robust design problem.

4.2.1 Problem Statement – Changing Design Parameters

Consider a manufacturer of hand exercisers. The manufacturer has considerable demand for the hand exercisers. However, competitors are catching up and the manufacturer wishes to provide the customers with custom-made hand exercisers that suit different sections of the market. At the same time, the manufacturer is not sure if the product platform that he/she designs will continue to suit customer needs for a long period. Hence, he/she also wishes to have flexibility in designing the product platform. If there are changes in the design parameters of the hand exerciser, the platform must still achieve its objectives.

There are three key assumptions in the formulation of the customizable hand exerciser problem:

1. The demand of the hand exerciser is assumed to be constant. Since the objective of this example is to provide a method for designing platforms that are robust to changing design parameters, the example is not made unnecessarily complicated by considering uncertain demand. The uncertain demand was considered in the previous example of this chapter. Also, the market space is assumed to be fixed. Changing extents of marketplaces is explored in Chapter 5.

2. We do not discuss the design and manufacturing of the plastic handle and we focus only on the design of the torsion spring in the hand exerciser. The modeling of the exerciser considers the cost of material, tooling, processing and packaging involved.

3. The design of the product platform is achieved by scaling of the product parameters. Customization is not achieved changing product functionality.

The example of the hand exerciser helps to illustrate the above-mentioned method of designing product platforms for changing design parameters. In a hand exerciser, there are a number of parameters such as coil diameter D, height H of the exerciser that must satisfy customer specific requirements. Depending on the age, size, using style, portability requirements of the customer, the design parameters needed will be different. So after the development of a product platform, it must be possible to have some freedom in changing these parameters during the product life cycle without affecting the profitability of the platform. In this example, the platform is designed to make it robust to changes in the coil diameter of the exerciser. The cost objective of the platform needs to be unaffected by changes in this parameter.

4.2.2 Modeling of Hand Exerciser Product

We illustrate the design of a line of custom-made hand exercisers similar to the one shown in Figure 4.20. The total demand of the exercisers is assumed to be 5000 per year.
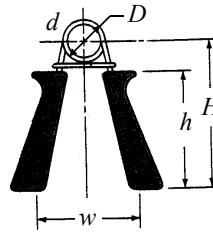
Figure 4.26: Sketch of a Hand Exerciser

It should be noted that this is a simplified example problem and the models used here do not represent those used in the actual design and production of these devices. This example has been developed in (Hernandez et al., 2003).

The manufacturing of the exerciser takes place in the following manner: First, a metallic wire is wound around a mandrel a certain number of turns. Then the ends of the wire are cut and a transition bend is made manually between the turns and the legs of the wire. Later on, the stresses in the wire are relieved in an oven. Then, the plastic handles are molded on, the grip squeezed together and a clip placed around the legs to bring the legs to the required initial position.

The exercisers are to be customized in the force F required to close the grip. F is calculated as in (Shigley and Mischke, 1996):

$$F = \frac{d^4 E \theta}{68 DNR}$$ [4.12]

Where d is the diameter of the wire, E is the elastic modulus of the wire material, D is the coil winding diameter, N is the number of coil turns, R is the distance from the coil circle center to the point where the concentrated force F is applied, and θ is the angular deflection from the no-load position to closing the grip. If we neglect changes in the coil diameter from the free position to the closed position, the angle θ can be approximated as:

$$\theta \approx \pi \left[1 - N + \text{int}(N) \right] - 2 \tan^{-1} \left(\frac{2H}{D} \right)$$ [4.13]

int(N) represents a function that returns the greatest integer $\leq N$. Substituting Equation (4.13) into Equation (4.12) yields:

$$F = \frac{d^4 E}{68 DNR} \left[\pi \left[1 - N + \text{int}(N) \right] - 2 \tan^{-1} \left(\frac{2H}{D} \right) \right]$$ [4.14]

The stress associated to the force F is approximated as:

$$\sigma = \frac{32 KFR}{\pi d^3}$$ [4.15]

where

$$K = \frac{4k^2 + k - 1}{4k(k-1)} \qquad [4.16]$$

and

$$k = \frac{D}{d} \qquad [4.17]$$

R is considered here constant and approximated as:

$$R \approx H - \frac{h}{2} \qquad [4.18]$$

We design the spring subject to the following geometric and structural constraints:

$$H \geq h + D \qquad [4.19]$$

$$\sigma \leq S_y \qquad [4.20]$$

$$\sigma\left(\frac{1}{S_e} + \frac{1}{S_u}\right) \leq 1 \qquad [4.21]$$

Where S_y, S_e and S_u are the yield, endurance and tensile strength of the wire.

The range of force considered in this example is:

$$10 \leq F \leq 110 \text{ Newtons} \qquad [4.22]$$

For this example we utilize values of *w*=90 mm and *h*=100 mm.

4.2.3 Product Platform Constructal Theory Method for the Design of a Hand Exerciser Platform to deal with Changing Design Parameters

In this section, the design of the hand exerciser platform using the augmented Product Platform Constructal Theory Method (PPCTM) is presented. Each of the eight steps discussed in Section 3.3 are explained using the example of a hand exerciser. The eight steps of the augmented PPCTM are presented in Figure 4.4.

Step 1: Define the Geometric Space and the Demand Scenario

The parameter that characterizes a hand exerciser is force that can be applied to it. The customizable spring is designed for 101 different forces: 10, 11, 12,…, 110 Newtons. This is a one-dimensional, discrete space of customization. The total annual demand of products is 5000 and we assume that all 101 forces have the same expected demand (5000 products per year for each force). Hence, the distribution of demand is uniform.

Step 2: Explore potential portfolio expansions

In Section 4.2.1, one of the assumptions noted is that there are no changes in the extents of the marketplaces in this example. Hence, there is no need of exploring potential portfolio expansions and so this step is skipped.

Step 3: Classify Design Parameters

This step differs in the pressure vessel example where one has demand as the noise factor. Here, the variation is present in one of the control factors. Hence, this is a Type 2 robust design problem. The coil-winding diameter D is one of the control factors to which variation is provided and its response (cost) is noted. The other values of control factors remain constant. The classification is illustrated in Figure 4.26.
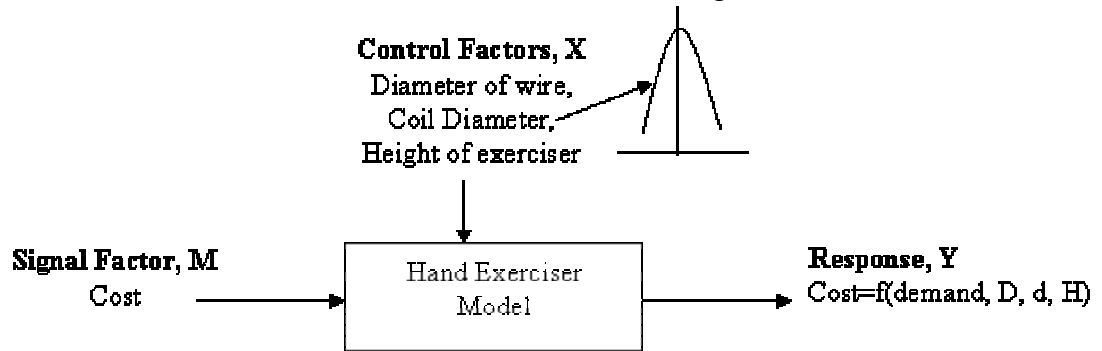
Figure 4.27: Classification of factors

Step 4: Define the Objective Functions

One of the objectives that are considered in this example is minimization of cost. Cost is typically formed by the costs of material, labor, tooling, inventory, packaging and others. Using similar analysis as in (Hernandez et al., 2003), we assume that most costs can be considered fixed or constant among spring variants and, therefore, do not need to be included in the optimization problem. The cost we consider is as follows:

$$Cost = Cmat + Ctool + Cproc + Cpack \qquad [4.23]$$

where *Cmat* is of the material used in the springs, *Ctool* is the cost of the mandrels, *Cproc* is the cost of processing associated with each wire used and *Cpack* is the cost of packaging of the exerciser. All these costs are estimated as total cost in a year.

The annual cost of material is estimated here as:

$$Cmat = \sum_{F} \delta(F) C_W \rho \left(\frac{\pi}{4} d^2 L(F,D,W,H) \right)$$

[4.24]

Where *F* represents each force of the space of customization, $\delta(F)$ is the annual demand of the spring used with *F*, C_W is the cost in dollars per kg of the required wire, ρ is the density in kg/m^3 of the wire, *d* is the wire diameter and *L(F,D,W,H)* is the required length of wire used in the spring as a function of the other design variables: *D*, the coil diameter, the wire selection *W*; and *H*, the distance from the bottom of the leg to the coil circle center (Figure 4.20). *L(F,D,W,H)* is approximated as:

$$L(F,D,W,H) = 1.2 \left[\pi D N(F,D,W,H) + 2 \left[\left(\frac{w-D}{2} \right)^2 + H^2 \right]^{1/2} \right]$$

[4.25]

In Equation (4.25) transition bend and other factors account for an extra 20% of wire. *N* is expressed in Equation (4.25) not as independent variable but as a function of *F, D, W* and *H* because *N* is always chosen as the *minimum* number of turns that satisfies Equation (4.12) for a given *F*:

$$N(F,D,W,H) = \min\{N(F,D,W,H)\} \qquad [4.26]$$

The wires available for selection are shown in Table 4.7.

Table 4.11: Available Wires

Wire	E [MPa]	ρ [kg/m^3]	d [mm]	S_y [MPa]	S_u [MPa]	S_e [KPa]	C [dollars/kg]
1	200x10³	7860	0.79	1978	2274	540	3.53
2	200x10³	7860	1.30	1816	2088	540	2.48
3	200x10³	7860	1.60	1756	2019	540	2.39
4	200x10³	7860	1.91	1702	1957	540	2.37
5	200x10³	7860	2.59	1618	1860	540	2.22
6	200x10³	7860	3.18	1565	1798	540	2.18
7	200x10³	7860	4.11	1493	1716	540	2.16
8	200x10³	7860	0.89	1678	1929	540	2.42
9	200x10³	7890	1.22	1559	1791	540	2.26
10	200x10³	7890	1.60	1499	1723	540	2.15
11	200x10³	7890	2.69	1349	1550	540	1.36
12	200x10³	7890	3.18	1319	1516	540	0.91
13	200x10³	7890	4.17	1139	1309	540	0.89
14	200x10³	7890	4.88	1330	1550	540	1.25
15	200x10³	7890	5.26	1310	1520	540	1.15

The cost of tooling *Ctool* is mainly determined by the number of mandrels m required. In order to determine *m* we make the following assumptions:

- A different mandrel is needed for each different coil diameter *D*.

- A mandrel is replaced every *Z* number of springs, where *Z*=50,000 for this example.

The maximum yield capacity *Y* per mandrel to maintain a utilization of 90% is 10 springs per hour, and the facility works 16 hours a day. Neglecting down time, *Y*=58,400 springs per year per mandrel.

With the assumptions noted, the annual cost of tooling is calculated as follows: Since all springs with the same coil diameter *D* can share the same mandrel, the total demand for each different diameter *D* is obtained as the sum of the demand for the various forces with springs with the same coil diameter:

$$\delta_D = \sum_{D(F)} \delta(F) \qquad\qquad [4.27]$$

The minimum number of mandrels needed at anytime, m_D, is the nearest integer higher than the ratio of δ_D to the yield capacity *Y* of a mandrel:

$$m_D = \mathrm{int}\left(\frac{\delta_D}{Y}\right) + 1 \qquad\qquad [4.28]$$

The total production per mandrel, p_D, is then:

$$p_D = \frac{\delta_D}{m_D}$$ [4.29]

Now, let us assume that the cost of a mandrel, $C_{mandrel}$, is the same for all diameters and equal to 1000 dollars. The cost per year of mandrels used to manufacture coils with diameter D, considering the need to replace mandrels every Z number of springs is then:

$$Ctool_D = C_{mandrel} m_D \frac{p_D}{Z}$$ [4.30]

In addition to the cost of Equation (4.30), we add to the tooling cost an opportunity cost due to the "non-ideal" utilization of equipment. We estimate this tooling opportunity cost as follows:

$$Ctool_{opp} = \sum_D r C_{mandrel} m_D \left(1 - \frac{p_D}{Y} \right)$$ [4.31]

where r is a annual interest rate (equal to 0.1 for this example). The ratio of p_D to Y yields the percent of time the mandrel is "idle" relative to the "ideal" production Y.

The total cost per year of tooling is then:

$$Ctool = \sum_{D} \left[C_{mandrel} m_D \frac{p_D}{Z} + rC_{mandrel} m_D \left(1 - \frac{p_D}{Y} \right) \right] \qquad [4.32]$$

The processing costs involve purchase processing, warehouse space and other factors associated with each wire used. Let us say that the extra cost is $2000 for each different type of wire used.

The cost of packaging involves the costs associated with packing each exerciser having different heights. This cost is assumed to be $350 for every different height of exerciser used. We do this to examine how our solution changes as the cost due to the height variety changes.

The total cost per year is then calculated by substituting Equations (4.24) and (4.32) in Equation (4.23). We proceed now to the third step of the method.

Step 5: Identify Modes for Managing Product Variety

Customization of the exercisers is achieved through the following modes for managing product variety:

1. Varying the distance H
2. Varying the number of turns N
3. Varying the coil diameter D
4. Varying the wire selection (i.e., the wire diameter d and its material properties)

Now we proceed to Step 6 to determine how many hierarchic levels to utilize and which modes should be used in which hierarchy level.

Step 6: Identify Number of Hierarchy Levels and Allocate the Modes for Managing Product Variety to These Levels

In this example, three hierarchic levels are used. The smallest divisions are referred to as *first-space divisions*, $\Delta F1$; the assembly of these as *second-space divisions*, $\Delta F2$; and, finally, the assembly of the later as *third-space division*, $\Delta F3$.

First Space Division

As discussed in Section 2.1, the lowest level of the hierarchy is comprised of the mode of managing variety that is most flexible and economical providing with high frequency and short-scope changes. For exercisers, this level needs to be fixed at manufacturing, unlike other products where flexible features can be provided. For a given w and h, it is economical and easy to vary the number of turns N and the distance H since no additional tooling or operations are required. At this level, each first space division has a common value of H between all products and adjusting the number of turns satisfies the required customization force of each exerciser contained in the space division.

Second Space Division

The second space element is composed by a number of first space elements. It has been shown in (Hernandez et al., 2003), that the variation of coil diameter provides better results than that of variation of wire diameter at this level. Hence, for this second space

185

element, commonalization of the coil diameter D is used to achieve variety with each first space division having its own value of H.

Third Space Division

The remaining mode for managing product variety available is varying the diameter of wire. We use this mode at the third level, i.e., highest hierarchy level, which is therefore allocated to the third space division, $\Delta F3$.

In summary, the space of customization is "accessed" through a hierarchic construct of modes for managing product variety as follows:

1. Varying the value of D at the third (largest) space divisions that contain a number of second space divisions.

2. Varying the wire selection at the second space divisions, which in turn contain a number of first space divisions.

3. Varying the height H at the first (smallest) space divisions.

4. Adjusting the number of coils for any specified force given the values of D, W and H.

A sketch of the hierarchic organization of modes for managing product variety is shown in Figure 4.28. Using this construct we proceed to formulate our problem mathematically in Step 7.
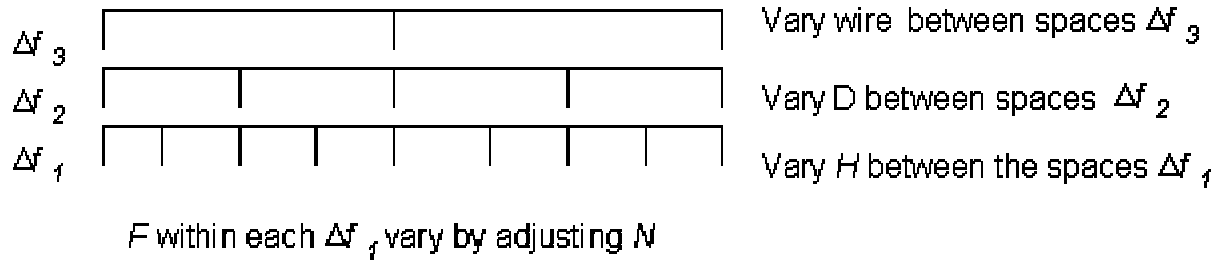
Figure 4.28 : Hierarchic Organization of the Modes for Managing Product Variety for Hand Exercisers

Step 7: Formulate a Multi-Stage Compromise Decision Support Problem

There are two broad level goals that need to be achieved for a robust design – "bringing the mean on target" and "minimizing deviation". In this problem, the former goal is to minimize the costs, while the latter goal is to minimize the variation in cost due to changing design parameters. Since there are two goals, a decision has to be made that gives the best possible combination of both. The compromise DSP is used to formulate this problem. The system goals are formulated for each decision stage of PPCTM because their focus is in achieving these two goals. The goal for each stage is based on the individual objective functions of cost and variance. The values of cost and variance to be used are found using average value and summation respectively, calculated in each decision stage.

In the formulation of the compromise Decision Support Problem, the goals are measured in terms of the deviation of the objective function from the ideal value. In this case, the minimum possible cost *IC* is divided by the actual cost *Cost* and the deviation of this ratio from the ideal value of 1 is calculated. Similarly, the minimum possible

variance *IV* is divided by the actual variance and the deviation of this ratio from the ideal value of 1 is calculated. Each set of objective function is represented by:

$$(IC / Cost) + d_{c1}^- - d_{c1}^+ = 1 \qquad\qquad [4.33]$$

$$(IV / Variance)] + d_{v1}^- - d_{v1}^+ = 1 \qquad\qquad [4.34]$$

The deviation variables are thus found. Using the weights of each objective as discussed above, along with the deviation variables forms the deviation function.

$$Z = W_1\, d_{c1}^- + W_2\, d_{v1}^-$$

The First Stage and the First Space Element

The decision variable of the first space element is Δf_i, the range of application of mode is to commonalize height H. The constraint is that the decision variable should be in the bounds of the force limits.

Decision 1: Δf_1

Given: Market Space of hand exerciser 10 - 110 Newtons

        Mode to commonalize height H

Find: The value of decision variable Δf_1

        The deviation variables, d_{c1}^- and d_{c1}^+, d_{v1}^- and d_{v1}^+

Satisfy: Bounds: $10 \le \Delta f_1 \le 110$ Newtons

    Constraints: $\sigma \le S_y$

$$\sigma\left(\frac{1}{S_e} + \frac{1}{S_u}\right) \le 1$$

...Contd. d_{c1}^-, $d_{c1}^+ \geq 0$

d_{v1}^-, $d_{v1}^+ \geq 0$

d_{c1}^-. $d_{c1}^+ = 0$

d_{v1}^-. $d_{v1}^+ = 0$

Goals: $(IC / Cost) + d_{c1}^- - d_{c1}^+ = 1$ [4.35]

$(IV / Variance)] + d_{v1}^- - d_{v1}^+ = 1$ [4.36]

Minimize: $Z = W_1 d_{c1}^- + W_2 d_{v1}^-$

where: *Cost* is calculated as in [4.23] for current market space

IC = Minimum cost for a year

V*ariance* = Variance of cost due to changes in D

IV = Minimum variance of cost

Figure 4.29: Formulation of the Multistage Compromise Decision Support Problem for the Hand Exerciser; First Stage Element

The Second Stage and the Second Space Element

The decision variable of the second stage is $\mathit{\Delta f_2}$. The range of application of the mode is to commonalize coil diameter D. One of the decision variable constraints is that it should be greater than the first stage decision variable $\mathit{\Delta f_1}$ or the lower limit of force and lesser than the upper limit of force.

Decision 2: Δf_2

Given: Market Space of hand exerciser 10 - 110 Newtons

   Mode to commonalize coil diameter D

   Value of Δf_1 from first stage

Find: The value of decision variable Δf_2

   The deviation variables, d_{c2}^- and d_{c2}^+, d_{v2}^- and d_{v2}^+

Satisfy: Bounds: $10 \le \Delta F_2 \le 110$ Newtons

   Constraints: $\Delta f_1 \le \Delta f_2 \le 110$ Newtons

$$\sigma \le S_y$$

$$\sigma \left(\frac{1}{S_e} + \frac{1}{S_u} \right) \le 1$$

$$d_{c2}^-, d_{c2}^+ \ge 0, \; d_{v2}^-, \; d_{v2}^+ \ge 0$$

$$d_{c2}^- \cdot d_{c2}^+ = 0, \; d_{v2}^- \cdot d_{v2}^+ = 0$$

   Goals: $(IC / Cost) + d_{c2}^- - d_{c2}^+ = 1$ [4.37]

   $(IV / Variance)] + d_{v2}^- - d_{v2}^+ = 1$ [4.38]

Minimize: $Z = W_{c2} \, d_{c2}^- + W_2 \, d_{v2}^-$

   where: *Cost* is calculated as in [4.23] for current market space

   IC = Minimum cost for a year

   V*ariance* = Variance of cost due to changes in D

   IV = Minimum variance of cost

Figure 4.30: Formulation of the Multistage Compromise Decision Support Problem for the Hand Exerciser; Second Stage Element

190

The Third Stage and the Third Space Element

The decision variable in this stage is Δf_3. The range of application of the mode is to commonalize the wire diameter.

Decision 3: ΔF_3

Given: Market Space of hand exerciser 10 - 110 Newtons

        Different types of wires available (Table 4.11)

        Value of Δf_2 from second stage

Find: The value of decision variable Δf_3

        The deviation variables, d_{c3}^- and d_{c3}^+, d_{v3}^- and d_{v3}^+

Satisfy: Bounds: $10 \leq \Delta f_3 \leq 110$ Newtons

    Constraints: $\Delta f_2 \leq \Delta f_3 \leq 110$

$$\sigma \leq S_y, \qquad \sigma\left(\frac{1}{S_e} + \frac{1}{S_u}\right) \leq 1$$

$$d_{c3}^-, d_{c3}^+ \geq 0, \ \ d_{v3}^-, d_{v3}^+ \geq 0, \ \ d_{c3}^- . d_{c3}^+ = 0, \ d_{v3}^- . d_{v3}^+ = 0$$

        Goals: $(IC / Cost) + d_{c3}^- - d_{c3}^+ = 1$ [4.39]

                  $(IV / Variance)] + d_{v3}^- - d_{v3}^+ = 1$ [4.40]

Minimize: $Z = W_{c3} \, d_{c3}^- + W_2 \, d_{v3}^-$

         where: *Cost* is calculated as in [4.23] for current market space

                IC = Minimum cost for a year

                V*ariance* = Variance of cost due to changes in D

                IV = Minimum variance of cost

Figure 4.31: Formulation of the Multistage Compromise Decision Support Problem for the Hand Exerciser; Third Stage Element

Step 8: Solve the Multi-Stage Compromise Decision Support Problem

Similar to the pressure vessel example reasons, a discrete analysis is used here. In discrete analysis, the analysis is done on a discrete set of points in the space, which helps to approximate the entire space. For discretization of the space, the designer chooses a certain resolution. Considering this resolution, nodes are established in the space of customization and objective function at every node is calculated.

In discrete analysis, the analysis is done on a discrete set of points in the space which helps to approximate the entire space. For discretization of the space, the designer chooses a certain resolution. Considering this resolution, nodes are established in the space of customization and the objective function at every node is calculated. This method was also used in (Williams, 2003) and seems to be appropriate for such problems in which expressing the objective function in terms of the design variables is difficult due to consideration of multiple objectives and changing design parameters.

Now the formulation of an appropriate solution technique needs to be determined. Similar solution algorithm as in the pressure vessel example is used; its primary goal is the determination of the extent of application of the modes that provides the least deviation from the ideal value, from which the commonality of design parameters of the product platform is determined.

Our solution technique is illustrated in Figure 4.31 (modified from Williams, 2003). This solution method involves iterating through values of the modes of managing variety (Δf_1, Δf_2, Δf_3), establishing the dimensions of the sub-spaces, commonalizing the design

parameters (d, D and H) across each sub-space, evaluating the objective functions, and calculating and comparing the deviation function of each iteration. Specifically:

- A particular combination of sub-space element sizes is taken as input. For example,

Δf_1	5N
Δf_2	10N
Δf_3	20N

These values must satisfy the bounds of these variables mentioned in the three compromise DSPs. The input also includes demand scenario, geometric space (market space) and the objective functions (minimize cost, minimize variance).

- From the sizes of the sub-space elements, the bounds of each sub-space are established. The values of parameters to be commonalized are determined from the bounds of the sub-space. This involves calculation of the parameters from their respective modes. For example, using the bounds of 5, 10, 15…. of Δf_1 the corresponding values of H are determined. Similarly such values of other specifications are determined at each level. For each specification, one needs to consider the assumed ranges at the start to obtain the value. Different combinations of the standard wires available need to be generated and corresponding objective functions calculated.

- These parameters are then commonalized across the respective sub-spaces. Hence, for any value of force in that range, the parameter value determined at the bound is used. The objective function is determined for every variant considering the parameter values across each sub-space.

- For calculation of the variance in cost, the first step involves calculation of cost from the parameter values. Then variance is provided to coil winding diameter D. The cost is then calculated while keeping other parameters constant. From the cost obtained from different values of coil diameter, the variance in cost is calculated. Specific to this example problem, ±20% variation is provided to the values of D in four discrete levels (i.e., Different values of D = {0.8D, 0.9D, 1.1D, 1.2D}, where D is the coil diameter), this is illustrated in Figure 4.33. P is the corresponding value of the objective function with variance σ.

- After obtaining the variance in cost for each variant, it is added up to get the total variance. Similarly, the cost of each variant is summed across the space.

- From the values of variance and cost, one obtains the deviation function as the output of this algorithm.

- The value of the deviation function is returned from the analysis and is compared with previous output. After each feasible combination of ranges of the modes for managing variety is explored, the input that provides the smallest deviation from the goal is selected.

So at the end of this solution process, the input Δf_2, Δf_2 and Δf_2 that gives the least deviation function Z is the product platform that best achieves the objectives. From these inputs, one obtains the corresponding specifications of the product family as shown in Table 4.12. If the customer wants a hand exerciser having force 12N, from Table 4.12, it is seen that the company will have to manufacture a hand exerciser having d=d1mm,

D=D2mm, L=H3mm. One can observe that a hand exerciser of 15N is provided to the customer having 12N requirement.

Table 4.12: Interpretation of Δf_1, Δf_2 and Δf_3

Force(Δf_3=20N)	Force(Δf_1=5N)	Force(Δf_2=10N)	d (mm)	D (mm)	H (mm)
20	5	10	d1	D1	H1
	10				H2
	15	20		D2	H3
	20				H4
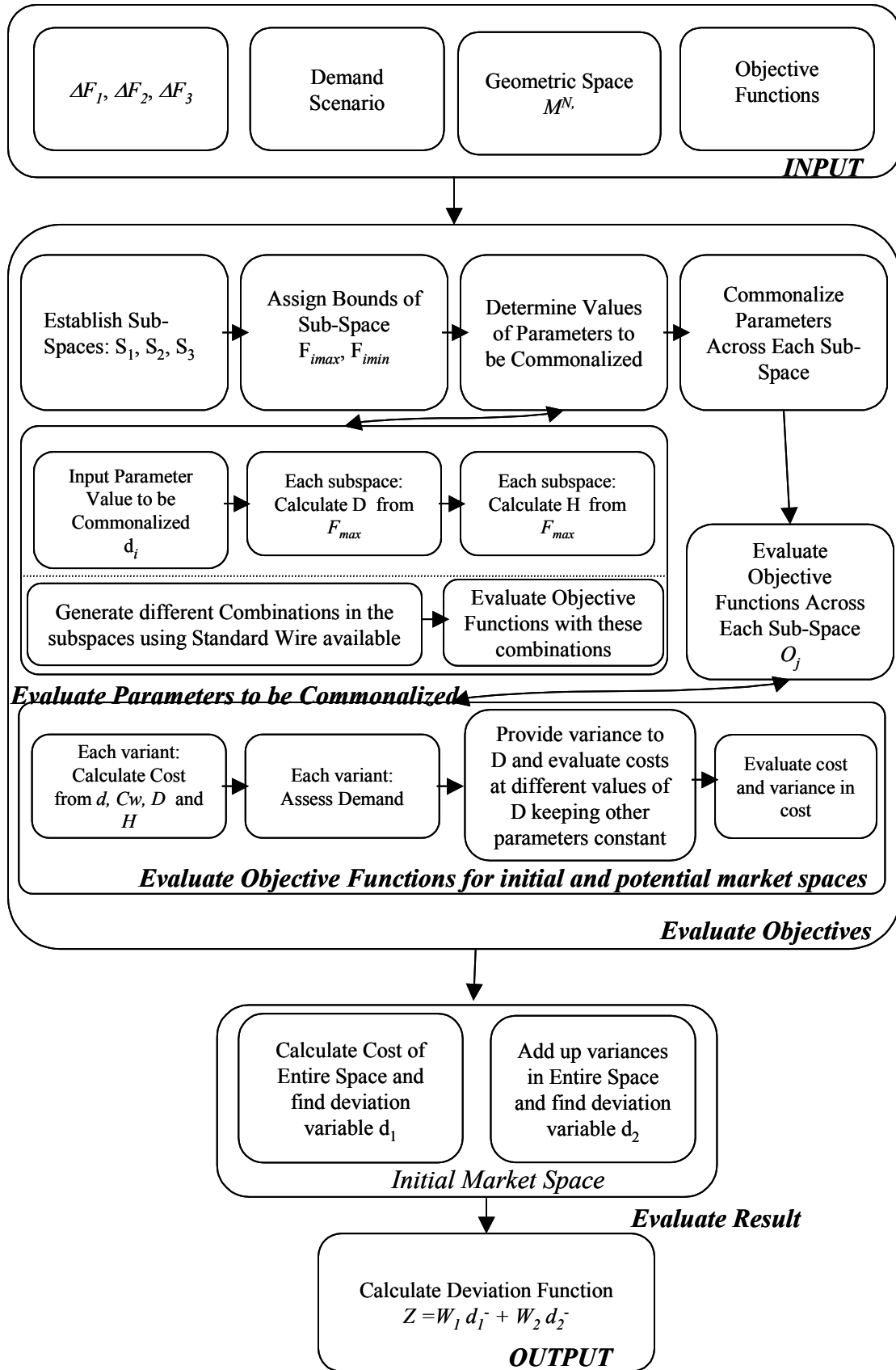
Figure 4.32: Solution Algorithm for Minimizing Variance in the Hand Exerciser example
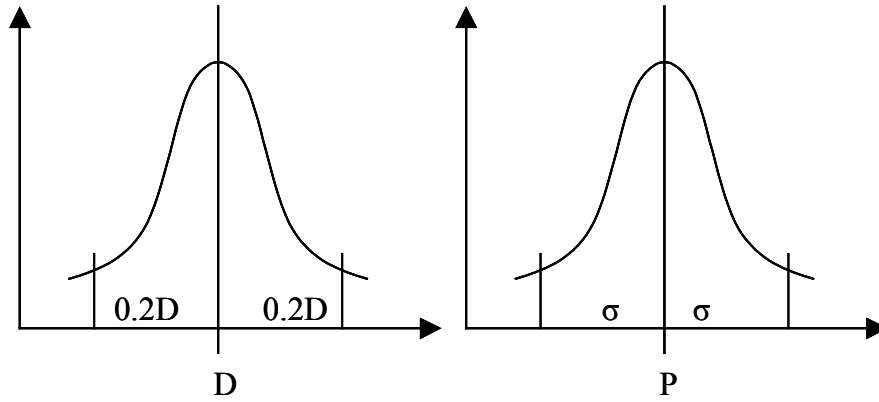
Figure 4.33: Coil Diameter and Cost – Mean and Standard Deviation

The solution to this problem is obtained by using optimization techniques or exhaustive search. First, the problem is solved with the exhaustive search method. Then optimization method is applied to obtain the solution and the solutions are compared.

Exhaustive Search

Due to reasons discussed in Section 3.3.8, an exhaustive search of different combinations of values of Δf_1, Δf_2, and Δf_3 is done. It is illustrated in Figure 4.34. However, the number of combinations is not restricted to these three variables. Since Δf_3 is for the determination of the type of wire, it is a decision to find an appropriate wire from Table 4.11 for particular sub spaces. So if Δf_3 sub space is 25N, there are 4 wires that need to be chosen and which wire will be used for which sub-space also needs to be decided. However, the large number of permutations is decreased by recognizing that the costs increase rapidly for smaller sub spaces of Δf_3. There are two reasons. The first reason is that more number of wires leads to more processing cost. The second reason is that having smaller sub spaces of Δf_3 leads to smaller sub-spaces of Δf_1 and Δf_2. This

197

increases the respective costs and there are no chances of getting a product platform with minimum cost with such combinations. It was observed that there are a maximum of 4 wires after which the costs increase drastically. Hence, one considers that there are 6 variables (Δf_1 and Δf_2 and 4 possible wires for Δf_3). There are more than 8000 such feasible combinations.

The input for the solution algorithm is different combinations of the ranges of the modes. After taking the ranges of the modes, the solution algorithm explained above calculates the value of the deviation function. This deviation function value is compared with the previous values. This is continued for different combinations of the ranges of the modes. The instantiation of this algorithm for the hand exerciser example problem is provided in Appendix C as a Java computer program.

The main benefit of using the exhaustive search is that one is able to explore all the possible combinations of the solutions. The number of computations is manageable and hence exploration of the entire space outweighs the computation intensity of the method. Moreover, it has been observed in (Williams, 2003) that this method gives better results than the other optimization techniques discussed.
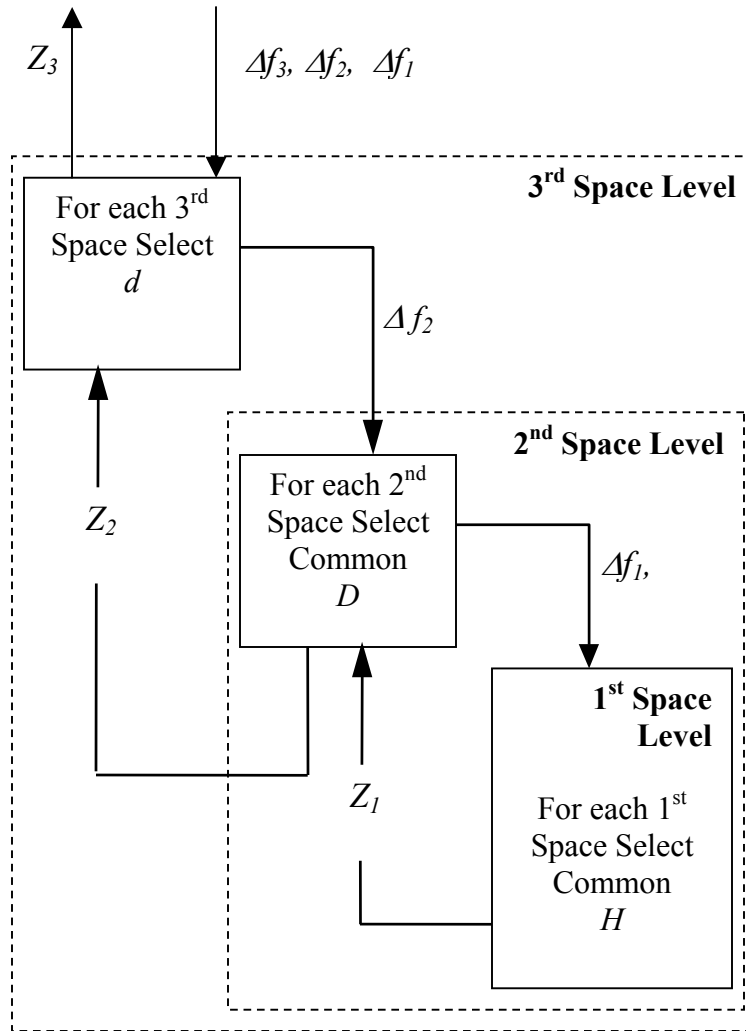
Figure 4.34: Exhaustive Search of Design Parameters for the Hand Exerciser example (Modified from Hernandez, 2003)

With the formulation of the PPCTM completed and its solution algorithm outlined, the results are presented in Section 4.2.4.

4.2.4 Results and Analysis – Scenarios in the Hand Exerciser Problem

Table 4.13: Section of the Results

$\Delta f3$ (N)	Δf_2 (N)	$\Delta f1$ (N)	Cost ($) Material	Tools	Processing	Packaging	Total Cost	Variance	Z
25 (5 6 13 6)	1	1	72337	9571	4000	35000	120909	3.63E+10	0.873
50 (6 5)	25	1	74673	5171	2000	35000	116845	3.91E+10	0.869
50 (6 5)	25	25	90779	5171	2000	1400	99351	7.99E+10	0.851
25 (6 5 6 13)	25	5	141220	5171	4000	7000	157392	1.88E+11	0.906
100 (6)	50	1	52422	5071	0	35000	92494	1.65E+10	0.828
100 (6)	50	10	56627	5071	0	3500	65199	3.01E+10	0.769
100 (14)	25	5	113534	5171	0	7000	125706	1.22E+11	0.882

Table 4.13 above shows a section of the results. Cost involves four terms – material, tools, processing and packaging costs. All these terms have been explained in the modeling of the hand exerciser in Section 4.2.2. In the end, the total cost and the variance in cost due to changes in coil diameter is determined. The objective function Z is then calculated from these two goals.

One observes that changes in $\Delta f1$ mainly affect the packaging costs. The reason is that more variety in heights of the exerciser leads to more cost of packaging. At the same time, it must be noted that each of the hierarchical stages affect the cost of material. It is obvious that more the commonality more is the material used and more is the costs.

The change in $\Delta f2$ mainly affects the tooling costs. Different types of diameters lead to the cost of purchasing different sizes of mandrels and hence tooling costs increases. At the same time, more commonality in the value of the diameter will lead to increase in material costs of the exerciser. All these effects are observed in the results in Table 4.13.

The column of $\Delta f3$ is comprised of different types of wires used for that iteration. The number in the brackets denotes the type of wire listed in Table 4.11. The change in

Δf3 mainly affects the processing costs. It is seen that more the different types of wires used, more is the cost of processing.

Three scenarios, similar to the pressure vessel example are run. In the first scenario, the design parameters are assumed to be constant throughout the product platform life cycle. In the second scenario, it is expected that the customer requirements and hence design parameters will vary. The third scenario is similar to the second scenario; however, for comparison purposes, the only objective that is considered is reduction in the variance of the objective function.

Scenario 1 – Design Parameters Constant

In this scenario, the design parameters are assumed to be constant throughout the product platform life cycle. Hence the objective of variance of costs is not considered, as there is no variance in the design parameters. The weight for the variance objective is zero while for reduction of costs is one. This is similar to the scenario in (Hernandez et al., 2003). As expected, the results show that this scenario has high variance and low costs. A detailed interpretation of the product platform is given in Scenario 2.

Table 4.14: Scenario 1 Product Platform Results

Scenario #	$k_{variance}$	k_{cost}	Δf3 (N)	Δf$_2$ (N)	Δf1 (N)	Cost ($)	Total Variance (^{2})	Z
1	0	1	50	50	25	32726.27	5.57E+09	0.114

The cost of material increases as the commonality of the Δf1 increases. However, similarly to other scenarios, the increase in commonality reduces the cost of packaging considerably. Similar to Scenario 2, the cost of tooling also decreases due to the use of only two types of coil diameters. It is observed that the value of Δf3 remains same in all three cases. The reason is that there is a balance between the costs of processing

associated with changing diameters and changes in the material costs due to different diameters.

Scenario 2 – Changing Design Parameters – A

In this scenario, the design parameter (diameter) does not remain constant throughout the product platform life cycle but the requirements change. So the designer needs to take care that the objectives of the product platform continue to be achieved despite changes in the design parameter. Hence in this scenario, both the objectives are considered. It has equal weight for both the goals – reduction in costs and minimization of variance. Hence, this scenario tries to balance both the objectives.

Table 4.15: Scenario 1 Product Platform Results

Scenario #	$k_{variance}$	k_{cost}	$\Delta f3$ (N)	Δf_2 (N)	$\Delta f1$ (N)	Cost ($)	Total Variance (^{2})	Z
2	0.5	0.5	50	50	10	32940.95	4.43E+09	0.503
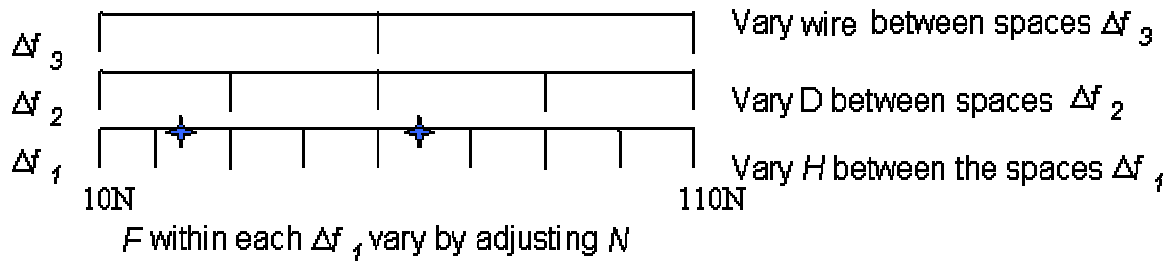
Figure 4.35: Hierarchical Arrangement of the Product Platform

Table 4.16: Section of the Scenario 2 Product Platform Specifications (LHS: 15N, RHS: 55N)

A				B		
d (mm)	D (mm)	H (mm)		d (mm)	D (mm)	H (mm)
		51.70985				51.70985
		79.85674				79.85674
5.26	52.7024	107.5559		5.26	52.7024	107.5559
		135.0875				135.0875
		162.5431				162.5431
		285.84				285.84
		326.9416				326.9416
3.18	34.8951	368.0615		3.18	34.8951	368.0615
		409.2043				409.2043
		450.3725				450.3725

The results inform the manufacturer that the best configuration of the modes of managing variety which closely matches his/her preferences is to commonalize the force at the lowest level for every 10N, at the second level for every 50N and the third level for every 50N.

Here, two customer examples are shown.

- Consider a customer having force requirements of a hand exerciser as 15N (denoted by left star in Figure 4.35). Using the different modes of managing variety at each hierarchical level (Figure 4.35), one obtains the product platform specifications involving d, D and H from the particular values of $\Delta f1$, $\Delta f2$ and $\Delta f3$. Design specification of the product family of this scenario is presented in LHS of Table 4.16. It is observed that the figures in bold in Table 4.16 satisfy the requirements of the customer. The hand exerciser of the product family that will satisfy the customer requirements is the one having wire diameter as 3.18mm, coil diameter as 34.89mm and height as 285.84mm.

- Consider another customer having force requirements of a hand exerciser as 55N (denoted by right star in Figure 4.35). The design specification of the product

family of this scenario is presented in RHS of Table 4.16. It is observed that the figures in bold in Table 4.16 satisfy the requirements of the customer. The hand exerciser of the product family that will satisfy the customer requirements is the one having wire diameter as 5.26mm, coil diameter as 52.7mm and height as 79.85mm

The cost of packaging decreases considerably due to larger value of $\Delta f1$. At the same time, as the commonality of height increases, the cost of materials increases but not to the same extent as the cost of packaging. Different types of diameters lead to the cost of purchasing different sizes of mandrels and hence more variety is not preferred. At the same time, more commonality in the value of the diameter will lead to increase in material costs of the exerciser. The weight on the variance is in between Scenario 1 and 3 and so the variance obtained in this case lies in between these two values.

Scenario 3 – Changing Design Parameters - B

The third scenario is similar to the second scenario; however, for comparison purposes, the only objective that is considered is reduction in the variance of the objective function. The third scenario has no importance for reducing costs. Hence it has low variance and high costs.

Table 4.17: Scenario 3 Product Platform Results

Scenario #	$k_{variance}$	k_{cost}	$\Delta f3$ (N)	Δf_2 (N)	$\Delta f1$ (N)	Cost ($)	Total Variance (^{2})	Z
3	1	0	50	25	1	65839.38	1.69E+09	0.703

It is observed from the small value of $\Delta f1$ that this product platform is comprised of a number of products, providing a large amount of variety in height of the exerciser.

204

Although this leads to a low value of cost of the material, the costs associated with the packaging increases much more significantly. Moreover, the costs of tooling associated with the use of different mandrels due to varying coil diameters also increases. Hence the total cost is considerably high. In this case, since the only objective is to reduce the variance in the costs, the reduction of the total costs is not considered and so the platform with the value of variance closest to the ideal value is found, irrespective of the cost associated with that platform.

Apart from the above discussed scenarios, the sensitivity of the results to different weights is studied. Scenarios are first run in which the weight is in between 0 and 0.5 for minimizing cost and in between 0.5 and 1 for minimizing variance. The Scenario 4 is seen in Table 4.18 where the weight for minimizing costs is 0.25 and weight for minimizing variance is 0.75. It is observed that the cost incurred ($35877.97) is less than the cost incurred ($65839.38) in Scenario 3 and variance (3.75E+09) is more than the variance obtained (1.69E+09) in Scenario 3. Also, the cost incurred ($35877.97) is more than the cost incurred ($32940.95) in Scenario 2 and variance (3.75E+09) is more than the variance obtained (4.43E+09) in Scenario 2. The reason is that the weight in this scenario is in between that of Scenario 2 and Scenario 3.

Table 4.18: Scenario 4 Product Platform Results

Scenario #	$k_{variance}$	k_{cost}	$\Delta f3$ (N)	Δf_2 (N)	$\Delta f1$ (N)	Cost ($)	Total Variance (^{2})	Z
4	0.75	0.25	50	50	5	35877.97	3.75E+09	0.698

The Scenario 5 is seen in Table 4.19. In this case the weight for minimizing cost is 0.15 and the weight for minimizing variance is 0.85. Here again, similar observations can be made as discussed earlier. This scenario falls in between Scenario 3 and Scenario 4.

Scenario #	$k_{variance}$	k_{cost}	$\Delta f3$ (N)	Δf_2 (N)	$\Delta f1$ (N)	Cost ($)	Total Variance (^{2})	Z
5	0.85	0.15	50	50	2	46063.51	2.48E+09	0.734

Another scenario is run in which the weight is in between 0.5 and 1 for minimizing cost and in between 0 and 0.5 for minimizing variance. In this case the product platform obtained is the same as in Scenario 2 for values of weights close to Scenario 2 while the product platform obtained is the same as in Scenario 1 for values of weights close to Scenario 1. The values of cost and variance in Scenario 1 and 2 are close to each other and hence change of weight does not lead to development of a new product platform. This can be better analyzed by considering the variation of the two objectives for different weights.
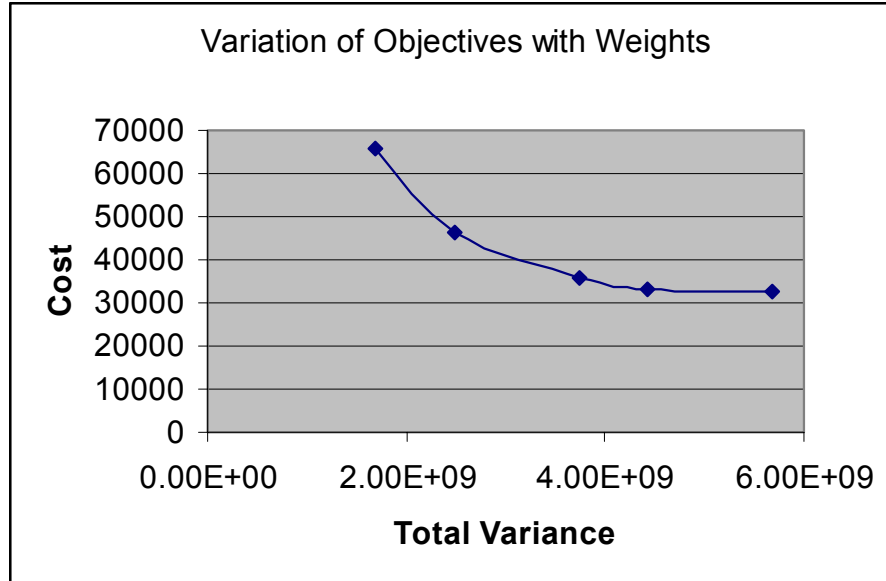
Figure 4.36: Plot of Objectives for Different Settings of Weights

The two objectives of minimizing cost and minimizing variance are plotted for different settings of the weights (Figure 4.36). It is observed that the slope of the line is greater in the range of weight 0 to 0.5 for minimizing cost while it is horizontal for 0.5 to 1 for same objective. This was observed in Scenario 1, 2, 3 and 4. There are more changes in the objectives in the range of 0 to 0.5 for minimizing cost than 0.5 to 1. So one can imagine that the manufacturer will tend to choose a platform having variance around 4E+09 since it gives him/her less penalty on cost while variance decreases by large amount. Based on this plot and his/her preferences, the manufacturer can choose the appropriate weights to design the product platform.

The type of wire chosen for each of the third stage element in every scenario is 6 and 15 as in Table 4.11. Now, instead of exhaustive search, the results are found using optimization techniques for comparison purposes. Different optimization techniques are used and their appropriateness for the problem discussed.

Optimization Techniques

Genetic Algorithms

Genetic algorithms, GA, (Goldberg, 1989) are inspired by Darwin's theory of evolution. Solution to a problem solved by genetic algorithms uses an evolutionary process (it is evolved).

Algorithm is started with a set of solutions (represented by chromosomes) called population. In the beginning a set of possible solutions are randomly generated. Solutions from one population are taken and used to form a new population. Solutions that are

selected to form new solutions (offspring) are selected according to their fitness - the more suitable they are the more chances they have to reproduce. Hence the chances of the new population to be better than the old one are higher. This is repeated until terminating condition (number of populations or improvement of the best solution) is satisfied. GA can handle real, integer and discrete types of variables.

In the chapter, Multi Island Genetic Algorithm (Abramson et al., 1992) is used. This divides the population into several islands. Then traditional genetic operations are performed on each island separately. Later, migration of individuals between the islands takes place. This helps to search many designs and multiple locations of the design space. Two parameters control the migration process – migration interval which is the number of generations between each migration, and migration rate which is the percentage of individuals migrated from each island at the time of migration.

In the method, each design point, i.e., every combination of $\Delta f1$, $\Delta f2$ and $\Delta f3$, is perceived as an individual with a certain value of fitness based on the value of objective function, i.e., value of Z, and constraint penalty. A combination with a better value of objective function has a higher fitness value. The selection operation in Multi-Island Genetic Algorithm uses the 'tournament selection' scheme (Goldberg et al., 1991).

Simulated Annealing

Simulated Annealing method (Kirkpatrick et al., 1983) has been used in optimization in a number of applications. The concept of this technique is based on the manner in which liquids freeze or metals recrystalize in the process of annealing. In an annealing process a melt, initially at high temperature and disordered, is slowly cooled so

208

that the system at any time is approximately in thermodynamic equilibrium. As cooling proceeds, the system becomes more ordered and approaches a "frozen" ground state at T=0. Hence the process is thought of as an adiabatic approach to the lowest energy state. If the initial temperature of the system is too low or cooling is done insufficiently slowly the system may become quenched forming defects or freezing out in metastable states (i.e., trapped in a local minimum energy state).

Optimization Techniques applied to the Problem

The different optimization methods in this section are implemented using iSIGHT (Engineous, 2003). Engineous' iSIGHT software enables rapid integration of programs and automates their execution to accelerate the exploration of design alternatives. In this project, it is used to automate the iterative cycle of inputting variables, performing a simulation program, getting the output, interpreting it and accordingly adjusting the input, and so on. This cycle is illustrated in Figure 4.37.
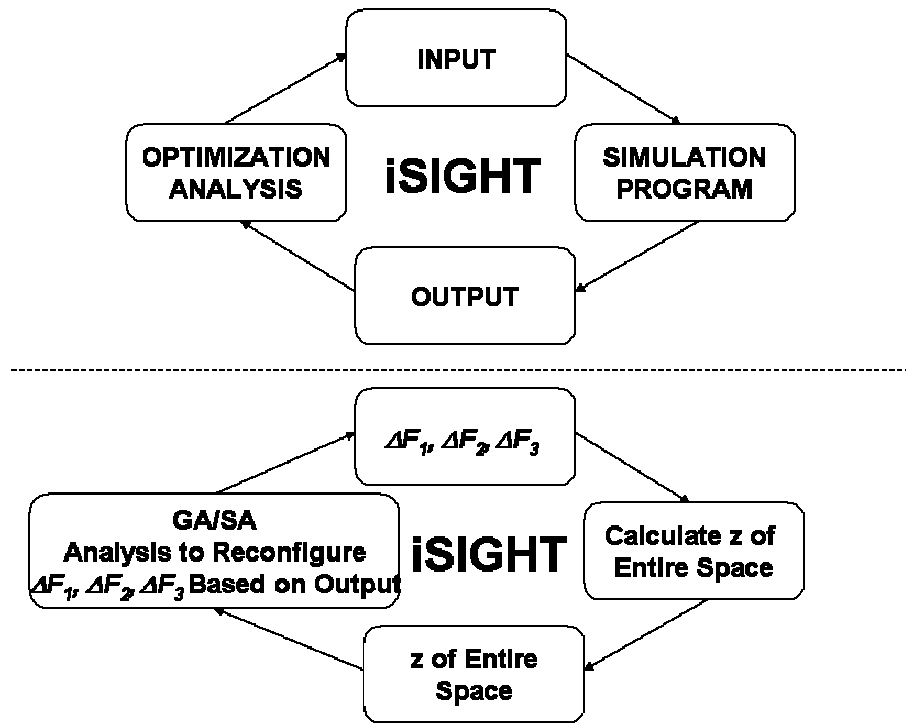
Figure 4.37: iSIGHT Automated Solution Cycle

As seen in the figure, $\Delta f1$, $\Delta f2$ and $\Delta f3$, which represent a product platform as explained in Section 4.3.7, are taken as input. For this particular platform (combination of $\Delta f1$, $\Delta f2$ and $\Delta f3$), the cost and variance is calculated for the entire space as shown in Section 4.3.3. The weighted average of the deviations is taken and this value Z determined needs to be minimized. Based on the analysis used and output obtained, $\Delta f1$, $\Delta f2$ and $\Delta f3$ are reconfigured and the process is repeated. The Java program for this is presented in Appendix C. For each of the optimization techniques, finding out the lowest value of Z is the sole objective. The product platform with the lowest possible value of Z will have the two goals of minimizing cost and variance best achieved.

In the proceeding sub sections, different optimization techniques are applied to the problem. In all cases, after solving the problem once, the output is taken as the

starting point and that particular optimization technique is used again to search for a better solution.

Genetic Algorithms - Simulated Annealing

In this case, genetic algorithms are used as the first step and then the optimum solution is obtained by using simulated annealing as the next step.

The specifications of the Multi-Island Genetic Algorithm are:

Table 4.20: Specifications of the Multi-Island Genetic Algorithm

Size of Sub Population	10
Number of Islands	10
Number of Generations	10

The specifications of the Adaptive Simulated Annealing algorithm are:

Table 4.21: Specifications of the Adaptive Simulated Annealing algorithm

Max. number of generated designs	10000
Number of Designs for Convergence Check	50
Convergence Epsilon	1.00E-08
Relative Rate of Parameter Annealing	1
Relative Rate of Cost Annealing	1
Relative Rate of Parameter Quenching	1
Relative Rate of Cost Quenching	1
Max. number of Failed designs	5

It is seen that as mentioned previously, the output obtained after the first run is taken as input for the next run to check if a better solution is obtained. It is seen in Table 4.22 that the value of Z reduces from 1 to 0.796466 in the first run. Later, further running of genetic algorithms is not able to achieve a lower Z.

The next step of simulated annealing is then applied. One observes that Z reduces to 0.5825895 in the first run. In the next run with this as starting point Z reduces to 0.512011. Later, further running of simulated annealing is not able to achieve a lower Z.

Thus, this is the lowest possible value of Z that can be achieved. The plot of deviation function versus the number of runs is shown in Figure 4.38.

Table 4.22: GA-Simulated Annealing Results 1

	Start	End
Genetic Algorithms	Δf3=100(15 15 15 15)	Δf3=50(11 11 13 13)
	Δf2=10	Δf2=50
	Δf1=5	Δf1=2
	Z=1	Z=0.796466
Simulated Annealing	Δf3=50(11 11 13 13)	Δf3=50(11 11 6 6)
	Δf2=50	Δf2=50
	Δf1=2	Δf1=25
	Z=0.796466	Z=0.583595
	Δf3=50(11 11 6 6)	Δf3=50(15 15 6 6)
	Δf2=50	Δf2=50
	Δf1=25	Δf1=25
	Z=0.583595	Z=0.512011

Figure 4.38: Deviation Function vs. Number of Runs 1

Second Case with Different Starting Point

In this case, again the same two steps are used with different starting points. It is seen in Table 4.23 that the lowest value of Z in this case is 0.659194. Thus one sees in

Table 4.23 that the method is not able to achieve the lowest value or close to the lowest value very regularly.

Table 4.23: GA-Simulated Annealing Results 2

	Start	End
Genetic Algorithms	$\Delta f3=100(1\ 1\ 1\ 1)$	$\Delta f3=50(4\ 4\ 2\ 2)$
	$\Delta f2=25$	$\Delta f2=25$
	$\Delta f1=1$	$\Delta f1=25$
	$Z=1$	$Z=0.730344$
Simulated Annealing	$\Delta f3=50(4\ 4\ 2\ 2)$	$\Delta f3=50(7\ 7\ 2\ 2)$
	$\Delta f2=25$	$\Delta f2=50$
	$\Delta f1=25$	$\Delta f1=10$
	$Z=0.730344$	$Z=0.659194$

Figure 4.39: Deviation Function vs. Number of Runs 2

There are a couple of lessons to be learned from this. First of all, both genetic algorithms and simulated annealing being exploratory techniques have the capability to search for solutions in the entire design space. It is observed that the method is robust enough to come closer to the actual solution, regardless of the starting point. Both the methods are

213

independent of the gradient. They are well-suited for discontinuous design spaces. Sequential Quadratic Programming is also used to solve the problem (Appendix D). Not being an exploratory technique it is able to obtain a good solution only if the starting point is close to the optimum. However, at the same time it is noted that the exploratory techniques are not able to obtain the product platform with the lowest value of Z, i.e., 0.5033 achieved by exhaustive search method. The author's aim is not to reduce the number of iterations and get a solution close to the best possible solution, but to obtain the product platform that will achieve the objectives in the best possible manner. The results from the optimization techniques are in line with the claims in (Williams, 2003). Similar differences in the results between optimization and exhaustive search method were obtained.

The author considers another discrete optimization technique Mixed Integer Programming combined with Genetic Algorithms to see if the product platform with the best possible achievement of objectives is obtained using it. A mixed-integer program is the minimization or maximization of a linear function subject to linear constraints. Mixed integer programs can be used to formulate just about any discrete optimization problem. They are heavily used in practice for solving problems in transportation and manufacturing: airline crew scheduling, vehicle routing, production planning, etc.

Mixed Integer Programming – Genetic Algorithms

In this case, a combination of Mixed Integer Programming (Belegundu, 1999) and Genetic Algorithms is used. It is seen in Table 4.24 that the value of Z is reduced from 0.891936 to 0.51201 in two runs. However, the number of iterations is more than that of

the combination of Genetic Algorithms and Simulated Annealing. Moreover, the platform obtained is close to the minimum and not *the* minimum.

Table 4.24: Mixed Integer Programming-GA Results

	Start	End
Mixed Integer Programming- Genetic Algorithms	Δf3=100(15 15 15 15)	Δf3=50(15 15 13 13)
	Δf2=50	Δf2=50
	Δf1=25	Δf1=25
	Z=0.891936	Z=0.717896
	Δf3=50(15 15 13 13)	Δf3=50(15 15 6 6)
	Δf2=50	Δf2=50
	Δf1=25	Δf1=25
	Z=0.717896	Z=0.51201

Figure 4.40: Deviation Function vs. Number of Runs

4.2.5 Economic Gains using the Method

Scenario A – Product platform without considering changing design parameters

Table 4.25: Cost Savings without Augmentations

Scenario	$k_{variance}$	k_{cost}	$\Delta f3$ (N)	Δf_2 (N)	$\Delta f1$ (N)	Cost ($)	Total Variance (^{2})	Z
A.1	0	1	50	50	25	32726.27	5.57E+09	0.114
A.2						33692		
					Difference	966		

Consider a manufacturer of hand exercisers who designs a product platform to minimize the costs associated with it. Since there is only one objective, the manufacturer does not consider shifts in customer requirements by changing design parameters. The resulting product platform is seen in Scenario A.1 of Table 4.25 along with the costs associated with it. This is the same scenario as in Scenario 1.

Now consider that customers' requirements have changed and they are looking for hand exercisers with a larger diameter, viz., 1.2 times the original diameter. Now if the manufacturer uses the existing product platform to incorporate these changes the costs associated with it are noted in Scenario A.2. The costs are obviously higher; however, the point to note is the amount of difference between the two costs. This is analyzed in more detail while considering the difference in costs of Scenario B.

Scenario B – Product platform considering changing design parameters

Table 4.26: Cost Savings with Augmentations

Scenario	$k_{variance}$	k_{cost}	$\Delta f3$ (N)	Δf_2 (N)	$\Delta f1$ (N)	Cost ($)	Total Variance (^{2})	Z
B.1	0.5	0.5	50	50	10	32940.95	4.43E+09	0.503
B.2						33752		
					Difference	812		

Now consider a manufacturer of hand exercisers who is aware of the changing customers' requirements and hence designs the product platform accordingly. The design of the product platform takes place by considering both the objectives of reducing the cost and reducing the variance in costs due to changing parameters. The resulting product platform is seen in Scenario B.1 of Table 4.26 along with the costs associated with it. This is the same scenario as in Scenario 2.

Again consider that customers' requirements have changed and they are looking for hand exercisers with a larger diameter viz., 1.2 times the original diameter. Now if the manufacturer uses the existing product platform to incorporate these changes the costs associated with it are noted in Scenario B.2. The costs are obviously higher; however the point to note is the amount of difference between the two costs.

Now one sees the difference between Scenario A and B. The difference in costs of Scenario B is less than that of difference in costs of Scenario A. Hence it is observed that if the product platform is designed considering the future changing customer requirements, the costs associated with using the existing product platform for the changed design parameter later on will be less. Thus the product platform in Scenario B

is robust to such changes in design parameters and will give favorable results despite of unforeseen changes in customer requirements.

The validity of results is established by four ways:

1. Checking the constraints

2. Comparison of a robust and non robust product platform

3. Values of objective function making small changes in design variables

4. Comparison of solution obtained by optimization techniques and exhaustive search

1. **Checking the constraints -** There are three constraints in the problem. Firstly, the range of modes at lower levels must be smaller than the range of modes at higher levels. This is seen from Table 4.27 that all the lower levels (e.g., Δf1) are smaller than the higher levels (e.g., Δf2). Secondly, the constraint that $d_i^-, d_i^+ \geq 0$ needs to be satisfied. Again, it is seen all the values in the Table 4.27 show that $d_i^-, d_i^+ \geq 0$. The third constraint is that $d_i^- \cdot d_i^+ = 0$. From the column of d+, it is seen that all the values are zero. Thus, one always obtains the product of d+ and d- as zero.

Table 4.27: Constraint Verification

			Cost		Variance	
$\Delta f3$ (N)	Δf_2 (N)	$\Delta f1$ (N)	d-	d+	d-	d+
25	1	1	0.7601	0	0.9862	0
50	25	1	0.7518	0	0.9872	0
50	50	1	0.6803	0	0.9634	0
25	25	1	0.7405	0	0.9838	0
50	25	5	0.7908	0	0.9965	0
25	25	25	0.6737	0	0.9915	0
50	50	10	0.7111	0	0.9932	0
25	25	5	0.3441	0	0.9231	0

2. **Comparison of a robust and non-robust product platform** – The Scenarios A and B help to validate the results. In these scenarios, one actually validates the results in Scenarios 1, 2 and 3. Scenario 1, 2 and 3 show that the reduction in variance in cost leads to less effect on cost due to changes in the coil diameter D. The Scenario A shows a non-robust product platform and the effect of having a different coil diameter on it is noted. Now the platform is made robust in Scenario B. Again the effect on the platform due to the use of some different coil diameter is determined. According to the model as shown in Scenario 1, 2 and 3, since the reduction in variance in cost leads to less effect on cost due to changes in the coil diameter, one should obtain results such that the robust product platform in Scenario 2 should have lesser change in cost than the non-robust product platform in Scenario 1. This is exactly what the results indicate.

3. **Values of objective function making small changes in design variables**

   The values of the decision variables are changed by a small extent from the solution obtained by exhaustive search and the change in the objective function is observed. It is seen that all the values of deviation function Z are higher than the obtained minimum.

   Table 4.28: Small Changes in Deviation Variables and its Effect on Deviation Function

$\Delta f3$ (N)	Δf_2 (N)	$\Delta f1$ (N)	Z
50(15 15 13 13)	50	10	0.71945
50(15 15 6 6)	50	25	0.51201
50(15 15 6 6)	50	5	0.52912

4. **Comparison of solution obtained by optimization techniques and exhaustive search**

The solution obtained by Multi-island Genetic Algorithms-Simulated Annealing (Table 4.22) and Mixed Integer Programming-Genetic Algorithms (Table 4.24) gave the same results with Z as 0.512011. The results of the optimization techniques were close to the exhaustive search solution having Z as 0.5033.

Some points from the discussion of the example problems of pressure vessel and hand exerciser are noted down.

Appropriate balance between objectives needs to be obtained according to situation

In the hand exerciser problem for changing design parameters, use of the product platform from the compromise case although increases the cost by a small amount ($32,940-$32,726) but decreases the difference shown in Scenario B ($966-$812) (Table 4.25 and Table 4.26). This difference indicating savings caused due to robustness to changes will overcome the increase in costs in a short amount of time.

However, one may argue if the chances of the design parameters to vary are less, then the platform is getting penalized by having to incur more costs. Here, the question of selecting a proper balance between the two objectives of bringing the mean to target and minimizing the variation arises. If the manufacturer knows that the design parameters variance is going to be very small, he/she can select appropriate weighing factors that will consider this situation. Hence, in such a situation, the platform will be penalized less however which will lead to fewer saving if there are changes of design parameters.

Consider the example of pressure vessel problem with uncertain demand. In this case there is a very small penalty ($4,698-$4,680) to the goal of bringing the mean to target. However, the difference in profits ($2,823-$2,816) due to significant variance in demand shown in Scenario 2 is also not very high (Table 4.8 and Table 4.9). So again one may argue if there are no such significant variances in demand in the future, why should one penalize the profit even by a small amount? In today's changing environment, the demand varies considerably and one would always try to handle such changing demand in the future. At the same time, as explained in the hand exerciser problem, one can vary the weighing factors to suit the appropriate problem at hand. So according to the problem if minimizing variation is a vital part of the problem, this objective will be given a high weight; however, if there is very less variation predicted in the future, this objective will be given a low weight and thus less penalty to costs.

Allocation of modes of managing variety in a hierarchical manner

Right now, there is no well defined systematic procedure for the allocation of modes of managing variety done in Step 5 of PPCTM-RCM. This has been missing in previous versions of PPCTM also. This is a limitation of the PPCTM. However, one can argue that although there is no such algorithm that is present for determining the right mode of managing variety at every level, it is possible to find the right allocation of modes of managing variety at each level. There are two steps to find the right allocation of modes.

1. As discussed in Step 4 of PPCTM in Chapter 2, the modes for managing product customization capable of achieving the smallest variations in design parameters

221

should be used first. Some approaches, such as dimensional customization, can be used to satisfy very small variations in the product, but are normally expensive for large changes. Other approaches, such as modularity, are not economically feasible for small changes. As such, economical and technological considerations play an important role in establishing the hierarchic use of the modes for managing product variety. In the example of pressure vessels, it is not possible to use the modular combination of vessels at a lower hierarchy since it is not economically feasible. Based on technological considerations, one can say that cutting different sheet lengths would cost less than having a different radius vessel and hence customization of length is at a lower level than commonalization of radius. However, one can argue that there could be modes of managing variety that are not as obvious in this case. For example, in the case of the hand exerciser, there is no such obvious reasoning to choose varying of coil diameter at a lower hierarchy than varying diameter of wire. In such situations, one needs to go to Step 2.

2. Consider that there is a doubt in the mind of the designer regarding allocation of two of the modes of managing variety. There are two possible hierarchies possible in such a situation. In this step, the designer should consider one such hierarchy and design a product platform based on it. Now, one should observe the subspace elements, $\Delta r(1),..., \Delta r~(i)$ (Figure 2.18). For instance, $\Delta r(2)$ and $\Delta r(3)$ are the subspace elements and the corresponding modes in question. If both the subspace elements are equal, ($\Delta r(2) = \Delta r(3)$), this means that the subspace element at the higher level is being restricted by the subspace element at the lower level. This is

checked by the constraint $\Delta r_j(i) \geq \Delta r_j(i-1)$ in the formulation of the compromise DSP at every stage. So in this case, the modes in the hierarchy need to be swapped and this would lead to a better product platform than the previous case. In the case of the hand exerciser example, similar analysis has been shown in (Hernandez et al., 2003). The variation of wire diameter is allocated at a lower hierarchy than variation of the coil diameter at the start. This hierarchical representation results in equal values of the subspace elements for the modes. Hence, swapping is done and the variation of coil diameter is put at a lower hierarchy than variation of wire diameter, which gives better results. Hence, in the hand exerciser example in this chapter also, similar hierarchy is used.

The author realizes that this is not an efficient way of finding out the allocation; however, it helps to get the right allocation in this manner. The way to address this problem is partly discussed in the *Future work* in Chapter 6.

Minimizing variance in objectives due to uncertain demand leads to less variety

In the pressure vessel example problem validation, it was noted that in Scenario 3 in Table 4.7 there are large values of each of the decision variables, $\Delta V_1, \Delta P_1, \Delta V_3, \Delta V_2$. After analyzing this in more detail, the reasoning was explained as follows. For instance, consider the case that the manufacturer goes for less commonality and more variety in such an uncertain demand. Now, it has been shown in (Williams, 2003) and logically one can say that large demand leads to more variety. However, if this demand is highly uncertain, more variety will lead to unnecessary losses for the manufacturer due to the possible decrease of demand in future. So the manufacturer would not opt for taking the

risk of providing a large product line but stick to fewer products. Hence the results make sense. However, one can argue that in the hand exerciser example no such trend is observed. The reason for it is as follows. In the hand exerciser example, variation is given to the design parameters and not the demand. Hence, it is a Type 2 robust design problem and not Type 1 where the noise is from outside. Moreover, the coil diameter to which the variation is given is a part of the hierarchy of modes of managing variety. So in determining the product platform in the hand exerciser example, the formulation of the example problem also needs to be considered. Hence, the result in Scenario 3 in the pressure vessel and hand exerciser problem cannot be compared.

4.3   A LOOK BACK AND A LOOK AHEAD

The purpose of this chapter along with Chapter 5 is to provide empirical structural and performance validity to the thesis. In Chapter 3, the augmented PPCTM is presented. The method is able to address the key issues of uncertain demand, changing design parameters and changing marketplaces. This chapter helps to focus on the former two issues. The augmented method presented in Section 3.3 is applied to the design of a line of customizable pressure vessels and hand exercisers.

The pressure vessel problem addresses the problem of uncertain demand in the design of product platforms. The pressure vessel is a classic mechanical engineering problem, which has been used for validations of the work of product platforms.

The hand exerciser problem addresses the problem of changing design parameters in the design of product platforms. It is a product that often requires customization. It is a challenge for manufacturers to achieve such customization economically due to the

competitive nature of the product. Customers are on the lookout for exercisers that exactly fit their requirements. So considering the importance of low cost to the customer as well as customized feel of the product, it is very important to be able to have a platform robust to any changes in the design parameters as per customer requirements. Therefore, this design problem is representative of a product that would benefit from the application of the theory and method proposed.

Hernandez and Williams have used the same examples in their previous work (Hernandez, 2003, Williams, 2003). Hence, it is useful to compare the results and analysis with previous methods. In each of the example problem, the Scenario 1 does not consider the augmentations suggested in this thesis. It is based on the method previously developed. Scenario 2 uses the augmented method PPCTM-RCM and shows the significance of the results compared to Scenario 1. If the original PPCTM is assumed to be a valid method, then the use of a similar example problem to test an augmented PPCTM is also valid.

Though the modeling of the problems looks relatively simple, the presence of a mix of continuous and discrete variables and non-linear functions and models make selection of design variables a complex task. At the same time, the main focus of the thesis is to concentrate on the method than concentrating on the complex modeling of examples. Hence, the selection of the example problems is relevant. The empirical structural and performance validity of the thesis (Figure 1.4) is partly provided in this chapter. The analysis and validity of the results have been discussed in Sections 4.2.5. It is shown that the method provides better results than a non-robust product platform design.

In Chapter 5, the issue of changing marketplaces is addressed by using the example of pressure vessels. It addresses the second part of the primary research question. The empirical structural and performance validity of the thesis is thus further examined in the forthcoming chapter.

CHAPTER V

DESIGN OF A FAMILY OF CUSTOMIZABLE PRESSURE VESSELS FOR A CHANGING MARKETPLACE

In Chapter 3, the augmented PPCTM is presented. This completes the theoretical structural validity of the thesis. In Chapter 4, the method is then applied to the design of a line of pressure vessel and hand exerciser problem to deal with uncertain demand and changing design parameters respectively. In this chapter, the augmented method is applied to handle changing extents of a pressure vessel market. This involves strategically designing a line of customizable pressure vessels that will consider objectives of current as well as potential future expansions of the market. This example helps to build on the method used to design a pressure vessel platform in the previous chapter, where there was fixed market space. As discussed in Section 2.4, strategic design involves taking action in accordance with the future trends well in advance. It is recognized in (Chamberlain, 2002) that the money invested by a company in a new product platform or in the development of a new piece of technology should continue to pay dividends when the next big market shift occurs a few months or years down the line. Strategic design, involving all such actions necessary for the company, plays a vital role.

In this section, the author augments the PPCTM with certain steps that helps design strategically. As discussed in Section 1.1.3, marketplace is the space in which the manufacturer has demand for his products and wishes to sell his products in this space.

This is the space in which the manufacturer designs his product family. The range of derivative products and associated customer requirements that are generated from a particular product platform is termed the *extent* of the product platform (Seepersad et al., 2002). However, in future the extent of the product platform may need to be expanded. So how would the manufacturer adjust to such changes in the marketplaces in the future? Based on concepts of strategic design, it would be right to take action in accordance with the future trends well in advance. Strategic design is meant to take those specific changes over time into account, forecasting trends and taking action in accordance with them well in advance. Chamberlain recognizes that the company should develop forecasts of future market trends to make sure that the money invested in a new product platform or in the development of a new piece of technology continues to pay dividends when the next big market shift occurs a few months or years down the line. The augmented steps in this chapter to PPCTM involve exploring potential portfolio expansions and considering its objectives, which help to take action in accordance with the future well in advance.

The author's primary objective in this chapter is to address the second section of the primary research question and hypothesis:

Q2 *"How should the Product Platform Constructal Theory Method be augmented so that platforms are designed strategically considering future portfolio expansions?"*

Hypothesis 2: Compromise DSP can be used in the Product Platform Constructal Theory Method to make tradeoffs between objectives of the initial market extent and future probable extensions.

The focus of this chapter is also to provide empirical structural and empirical performance validity of the augmented PPCTM. The positioning of this chapter in the thesis is understood from Figure 5.1 and Figure 5.2.
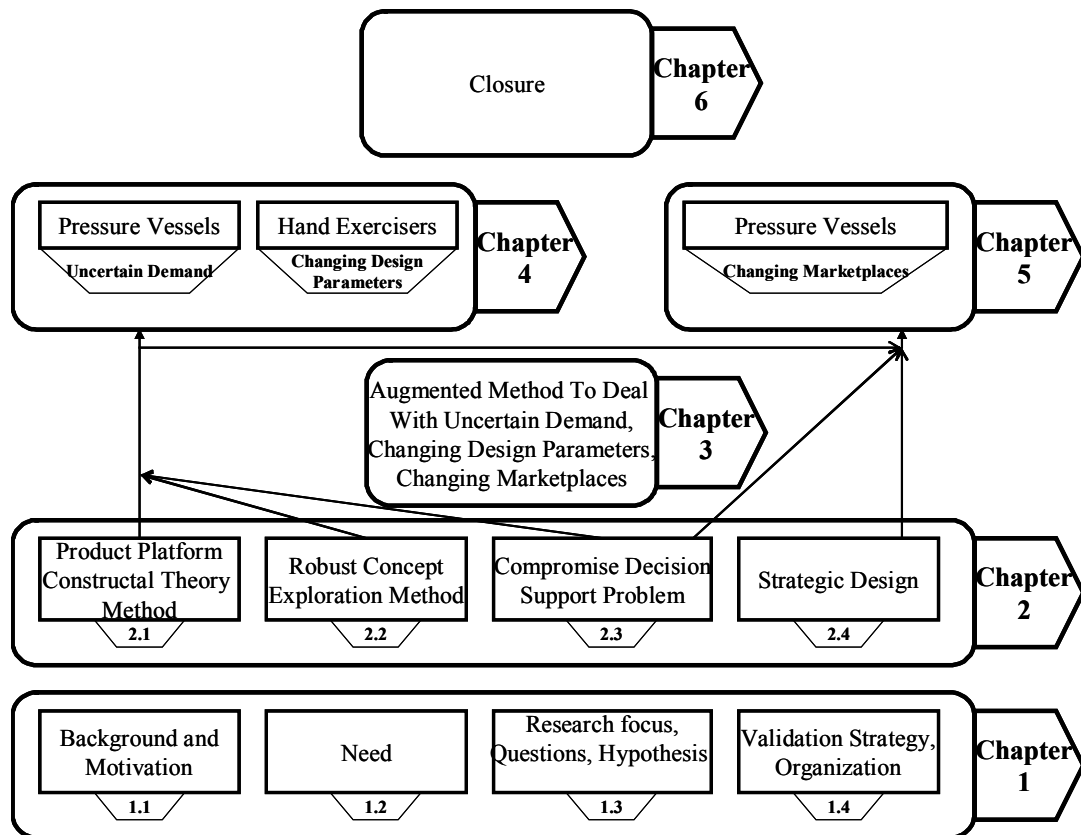
Figure 5.1: Overview of Chapters
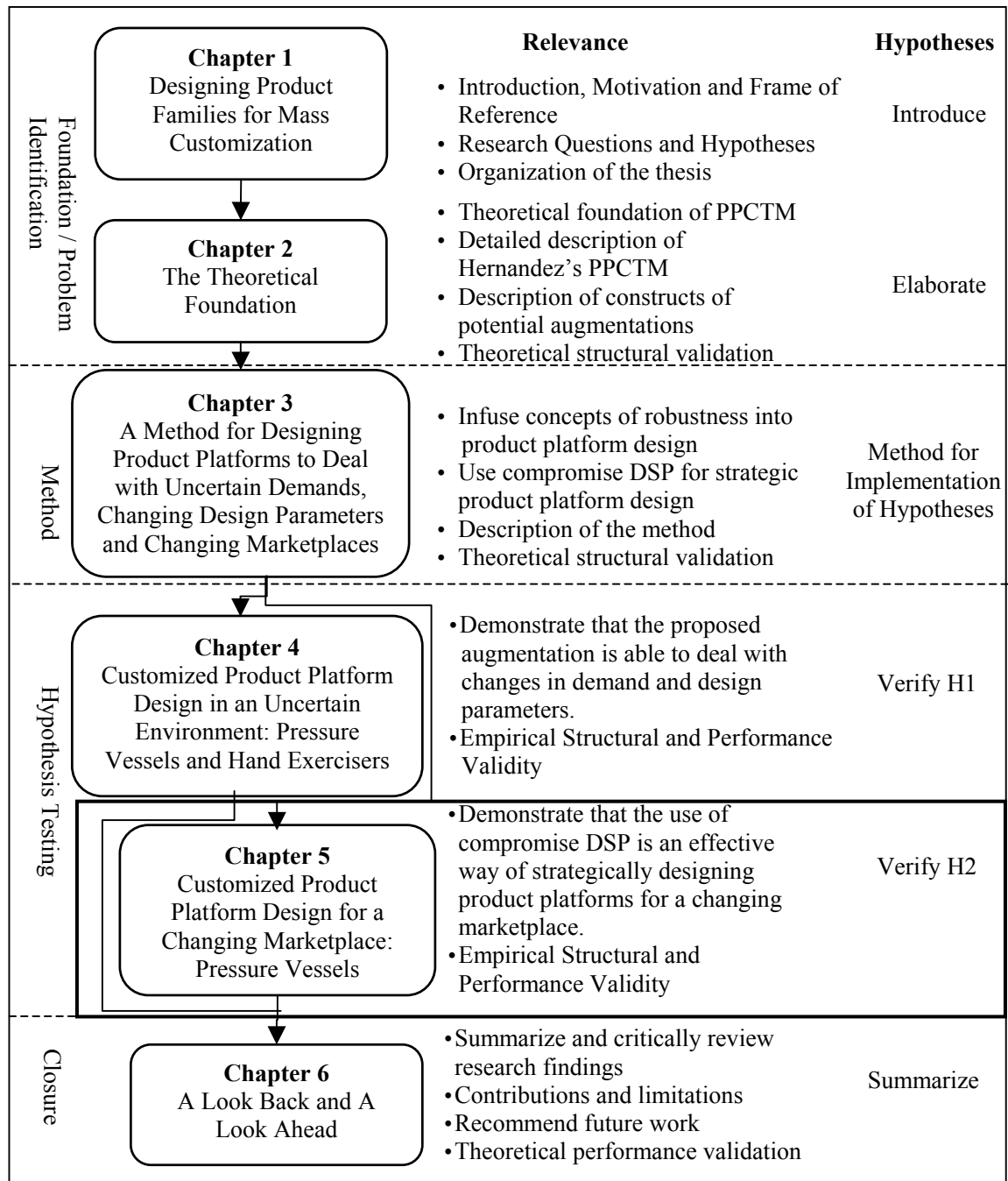
Figure 5.2: Thesis Roadmap

In Section 5.1, the problem statement is presented. In Section 5.2, some of the aspects of modeling of the pressure vessel have been discussed. Section 5.3 presents the

application of the augmented method to the problem in hand. Section 5.4 is comprised of the results and its analysis. Section 5.5 helps to validate the results. In the end, a look back and ahead is made.

This example problem of designing a family of customizable pressure vessels has been presented in previous applications of the PPCTM (Hernandez, 2001; Carone et al., 2003; Williams, 2003; Hernandez et al., 2003). As such, it is an appropriate example for the validation of the augmentations provided in this work since it is useful for comparison purposes.

5.1 PROBLEM STATEMENT – CHANGING EXTENTS OF PRESSURE VESSEL MARKET

A manufacturer of pressure vessels seeks a competitive advantage over other leading manufacturing enterprises through offering customized vessels. The manufacturer wishes to offer vessels that range in volume from 10 to 30 m^3, and in pressure from 10 to 30 MPa. By providing customers the opportunity to choose a pressure vessel suited to their specific needs, it is assumed that the resulting customized product family (and all of its variants) will be in high demand and therefore profitable. As is typical of markets of customizable products, the demand is highly fragmented and uncertain. The basic structure of the demand is assumed to be concentric square. As such, the manufacturer wishes to develop a product family that can efficiently offer customizable pressure vessels in a market of uncertain demand as well as a market having variable extents.

In this problem two scenarios are considered:

Scenario 1 – Design considering present market OR future market:

The first scenario has two cases – designing only for the present market and designing only for the future market. In the first case, we consider a manufacturer of pressure vessels having a market defined by volume in the range of 10-30 m^3 and pressure in the range of 10-30 MPa. This is the current market situation for the manufacturer. So the manufacturer designs a product platform using the original PPCTM in this marketplace.

Now consider that after a certain time interval, the demand for pressure vessels increases in the neighboring market. So the manufacturer wishes to extend his market space so that his new market includes the old as well as the new, high demand market space. In other words, the manufacturer wishes to have the market place having the pressure in the range of 10-40MPa. The manufacturer has two options.

Option 1: Now the problem that the manufacturer faces is that if the existing platform is extended to also include the new market, it will not continue to give favorable results. The reason is that the existing product platform is designed to achieve maximum profits in the range of 10-30MPa, so extending the same platform to give favorable results in the extended market is not possible.

Option 2: Redesigning the entire product platform for this extension of the market is not feasible. It will involve a lot of restructuring costs that makes it not possible. Thus, in this scenario, the results obtained for a manufacturer considering only the current market situation are analyzed.

Scenario 2 – Design considering present and future markets:

In the second scenario, while designing the initial product platform for the current market space, the manufacturer considers the possible future market condition. Considering the probability of different market situations in future, the manufacturer predicts the most probable market extension of the future. Here, let us assume that the demand will be higher in the range of 30-40MPa. So in this case, the manufacturer strategically designs the initial product platform using the augmented method proposed in this paper. This consideration helps the manufacturer not to fall into problems discussed in Scenario 1. Section 4.2 helps to solve the problem in second scenario.
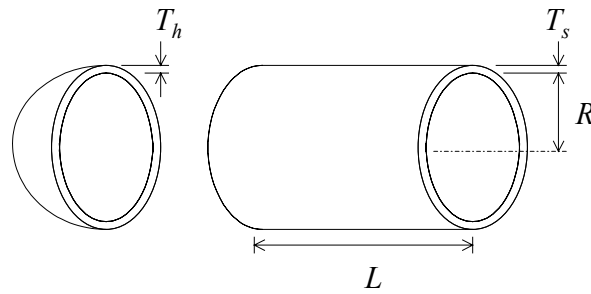
Figure 5.3: Schematic of Pressure Vessel Embodiment (Hernandez, 2001)

For this example, it is assumed that the conceptual design phase and the beginning of the embodiment phase of the pressure vessel have already been completed. Each vessel consists of a cylindrical container capped on both ends by hemispherical heads. A schematic of the basic design is presented in Figure 5.3.

The manufacturer wants to specify for each possible combination of pressure and volume the following design variables: length (L), radius (R), and the head and shell thickness (T_h and T_s). The manufacturer has determined that there are two objectives in

the design of the robust product platform: the maximization of the average profit, and the minimization of its deviance from changes in product demand.

The basic modeling of the pressure vessel example remains the same as in Section 4.1.2. Hence it is not repeated here.

5.2 STRATEGIC DESIGN OF A PRESSURE VESSEL PLATFORM TO HANDLE CHANGING EXTENTS OF MARKETPLACES

In this section, the different steps of the augmented PPCTM are applied to the pressure vessel example.

Step 1: Define the Geometric Space and the Demand Scenario

Product customization is offered in two independent design specifications – the volume and pressure of the vessel. The resulting two-dimensional continuous space of customization is shown in LHS of Figure 5.5. The demand in the market space can take any form – discrete, concentric square, linear continuous, gaussian distribution or any random model of demand. In this case, concentric square variation of demand is considered as shown in Figure 5.4.
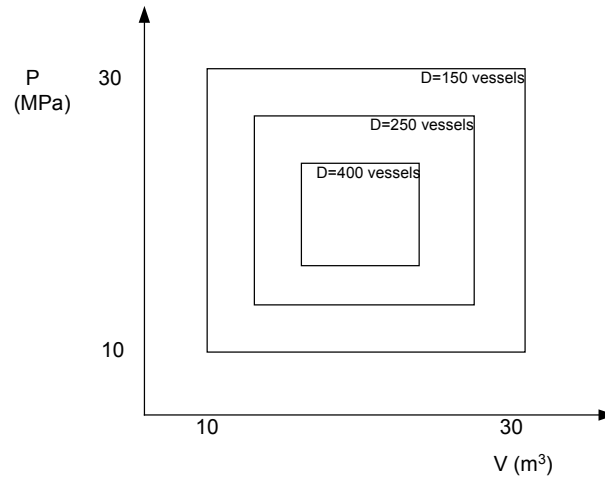
Figure 5.4: Pressure Vessel Space of Customization

Step 2: Explore potential portfolio expansions

In this step, the manufacturer needs to explore the different future potential markets for the pressure vessels. It is assumed that the manufacturer finds the market of pressure vessels will increase in the vertical direction. One also needs to consider for how long the market situation will remain in that manner and the probability of it to actually be profitable. It is assumed that the market will remain in the expanded position for a longer time and more profitable than the current situation and thus the manufacturer would want to extend it. Considering all such factors, the probability for the manufacturer of such a market expansion in the future is 80%. The extent of this potential expansion of the market is from 30MPa to 40MPa. After this extension, the manufacturer wishes to continue in the existing market space.

235
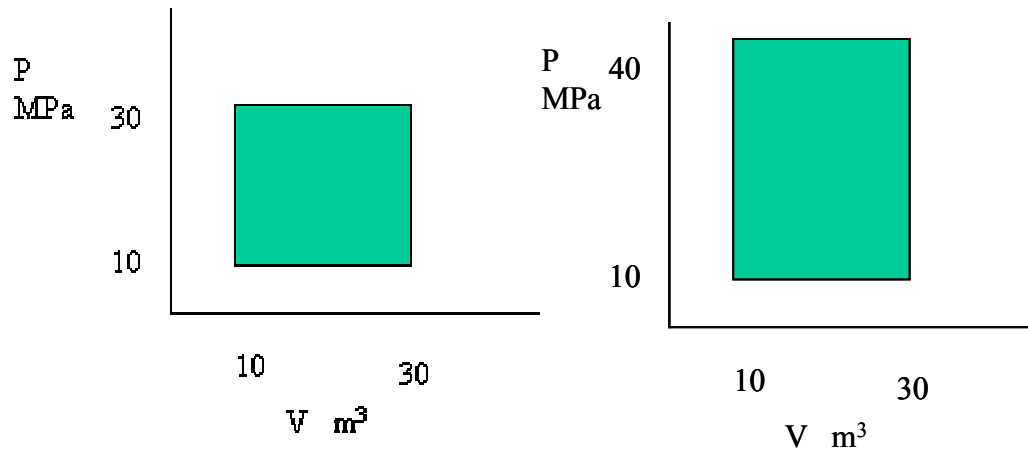
Figure 5.5: Initial and potential market extents

Step 3: Classify Design Parameters

This step begins the formulation of the problem by classifying the design factors. The classification is illustrated in Figure 5.6.
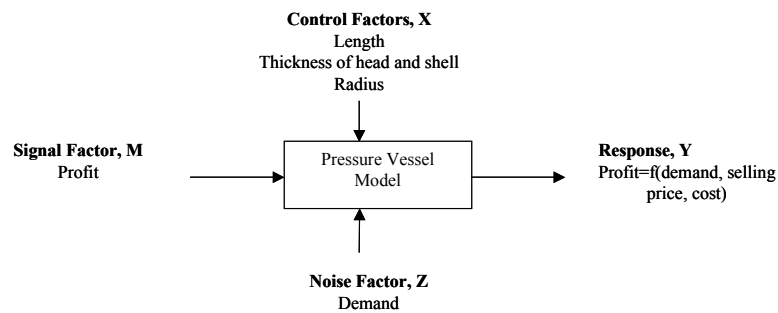
Figure 5.6: Factors and Response for Pressure Vessel Design

Step 4: Define the Objective Functions for Initial and Potential Product Portfolios

The manufacturer has four conflicting goals – two for the initial product platform and two for the future potential expansion. Here, the two objectives of each product platform are maximizing the profit and the other is to minimize the deviation in profit due

236

to fluctuations in demand. The primary objective, maximizing the profit, is the goal that the manufacturer wishes to "move the mean to target." From (Williams, 2003) the average profit of the market space is calculated as:

$$AvgProfit = \frac{1}{D_{tot}}\left[\left(\sum_{i=1}^{n} D_i(SellingPrice_i - C_i)\right) - C_{order} - C_{equip}\right] \qquad [5.1]$$

where, *SellingPrice_i* is the selling price of each product variant, C_i is the cost of each pressure vessel, C_{order} and C_{equip} are the costs associated with ordering and manufacturing equipment, respectively, and D_i is the demand of the specific product variant *i*, and D_{tot} is the total demand of the products of the market space (a detailed derivation is provided in the Section 4.1.2). As seen in Figure 5.6, demand is considered as the noise factor for this example. The other goal of this problem is to determine the control factors that minimize the variance of profit that typically arises from fluctuations in demand. The variance in profit is calculated by the following formula:

$$\sigma^2 = \sum (P_i - P)^2 / (n\text{-}1) \qquad [5.2]$$

where P_i is the profit calculated for demand D_i,

    P is the profit at mean demand, and

    n is the number of levels of the noise factor.

Considering the entire market space, the sum of variances of all considered products in the market space is calculated. The objective is to minimize the sum of variances for a particular product platform.

Step 5: Identify the Modes for Managing Variety

After brainstorming, designers have identified four different modes of managing product variety in pressure and volume.

Dimensional Customization of Shell Length (V1)

For this mode, the radius is kept constant while changes in volume are achieved by cutting each shell length from a stock piece of raw material. Hence, for a certain range of volumes, there is a common stock length of material with the radius remaining same. While it is cheaper to offer the minimum required length for a given radius, there is an ordering cost associated with each different length of raw material.

L_1 L_2

Figure 5.7: Dimensional Customization of the Length of the Shell (Hernandez, 2001)

Commonalization of Radius (V2)

The changes in volume in this mode are achieved by modifying the radius of a vessel. Since each different radius used leads to a new press and a die purchase, there is an increase in equipment cost. At the same time, it is obvious that there also exists a cost savings in material and waste costs by having radii that closely match that which is required.

Modular Combination of Vessels (V3)

In this mode, modular combination of vessels is used to achieve discrete changes in volume. This is done by combining vessels of two lesser volumes. This mode affects the costs of manufacturing equipment as well as the material costs. There is a saving on the purchase of manufacturing equipment costs while the material costs associated with using two smaller vessels may offset this savings. Hence, in this mode one determines the range of volumes for which modularity is used.

Figure 5.8: Modular Combination of Vessels (Hernandez, 2001)

Standardization of Thickness of Head and Shell (P1)

In this mode of managing variety in pressure, the modifications in the thickness of both the shell and the head of the vessel provide a way of changing the pressure. It is based on the fact that if a shell with dimensions R and L, and thickness T_s and T_h satisfy the pressure constraints for a pressure P_1, it also satisfies these constraints for any pressure $P \leq P_1$. There is also an additional charge for use of different thicknesses.

Step 6: Identify the Number of Hierarchy Levels and Allocate the Modes for Managing Variety to the Levels

In this step, the different modes of managing product variety are ordered hierarchically.

The First Stage and the First Space Element

As stated in Section 2.1, the first stage in the hierarchy is held by the mode that offers the smallest variations in the design parameters. Hence, out of the different modes available for managing variety, the mode with the highest fidelity is selected for this stage. For this first space element, Modes V1 (dimensional customization of shell length) and P1 (standardization of head and shell thickness) are used together to achieve variety in both volume and pressure. Dimensional customization of shell length makes small

variations in volume possible. By cutting the raw material, of length L_o, to any length desired, L, this mode is able to achieve continuous changes of volume.

The value of the variables ΔV_1 and ΔP_1 determines the size and shape of the first space element, S_1 (Figure 5.9). ΔV_1 and ΔP_1 represent the extent to which these modes of managing variety are applied to the platform. These need to be determined at this stage and are therefore the decision variables for this first stage.
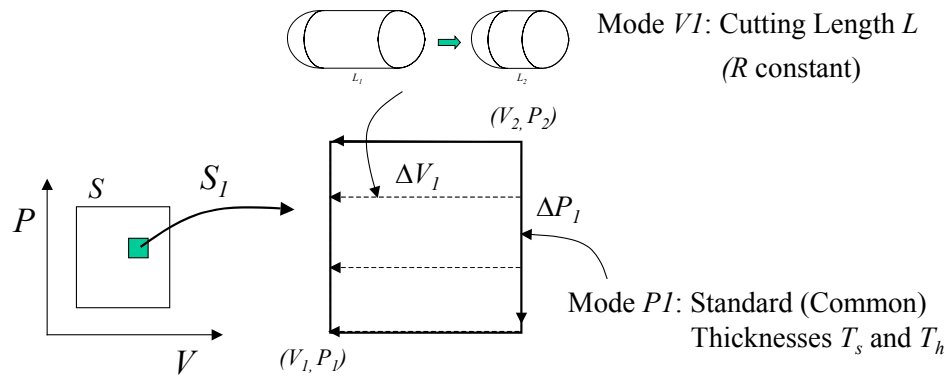
Figure 5.9: The First Space Element of the Pressure Vessel Space for Customization (Hernandez, 2001)

The Second Stage and the Second Space Element

The second space element, S_2, is composed by a number of first space elements, S_1, in the volume dimension. The mode after dimensional customization of shell length that can offer small variation in volume is the communalization of radius (Mode V2). Hence, for this second space element, commonalization of the radius (Mode V2) is used to achieve variety in volume. The value of the variable ΔV_2 determines the size and shape of the second space element, S_2 (Figure 5.10). ΔV_2 represents the extent to which the radius is commonalized across the platform and is therefore decision variable for this second stage.
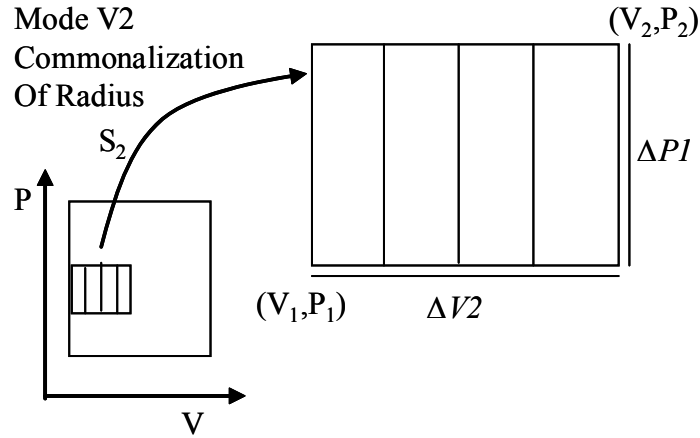
Figure 5.10: The Second Space Element of the Pressure Vessel Space for Customization

The Third Stage and the Third Space Element

The third space element, S_3, is composed of a number of second space elements, S_2, in the volume dimension. For this third space element, modularity of vessels (V3) is used to achieve variety in volume (Figure 5.11). The value of the variables ΔV_3 determines the size and shape of the third space element. ΔV_3 represents a point in the space in which modular combination of vessels begins, and is therefore decision variable for this third stage.

For example, if there exist customizable vessels in a range V_{min} to V', volumes larger than V' (up to $2V'$) are achieved by combining vessels within this range. In Figure 5.11, V' is 20 m^3, so all vessels with volumes greater than V' is achieved by combining two vessels. Hence, the variable to be determined in this case is the cut-off point (V') – a point where vessels with larger volumes are composed of two smaller vessels below the cut-off point.
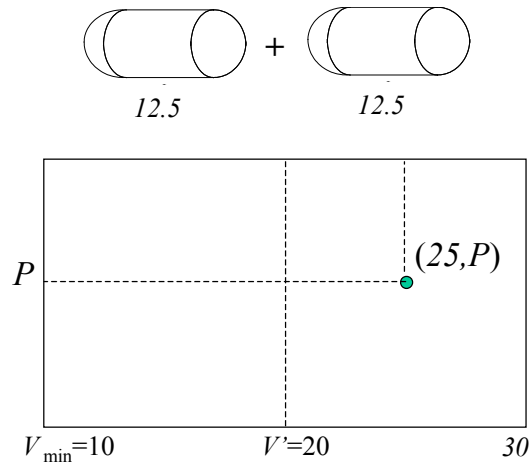
Figure 5.11: An Example of the Modular Combination of Vessels (Hernandez, 2001)

All the three subspaces with the decision variables are seen together in the Figure 5.12 below.
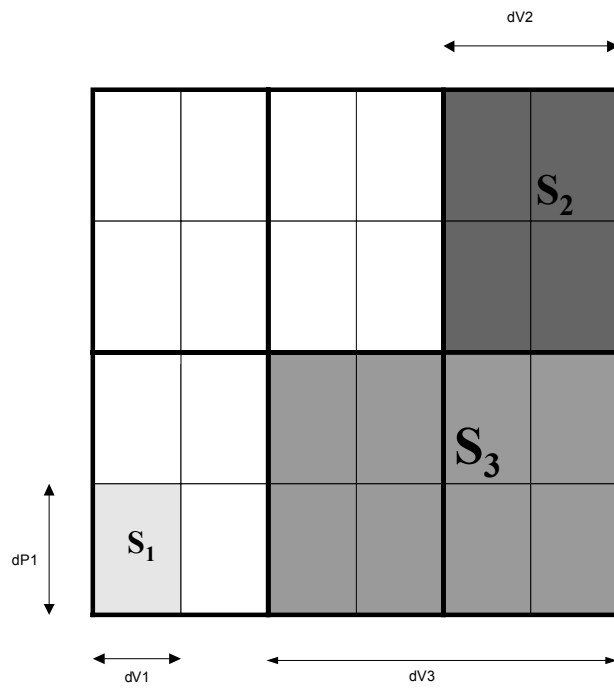
Figure 5.12: Product platform in terms of dV3, dV2, dP1, dV1

A graphical representation of the different modes of managing variety is seen in the Figure 5.13 below.
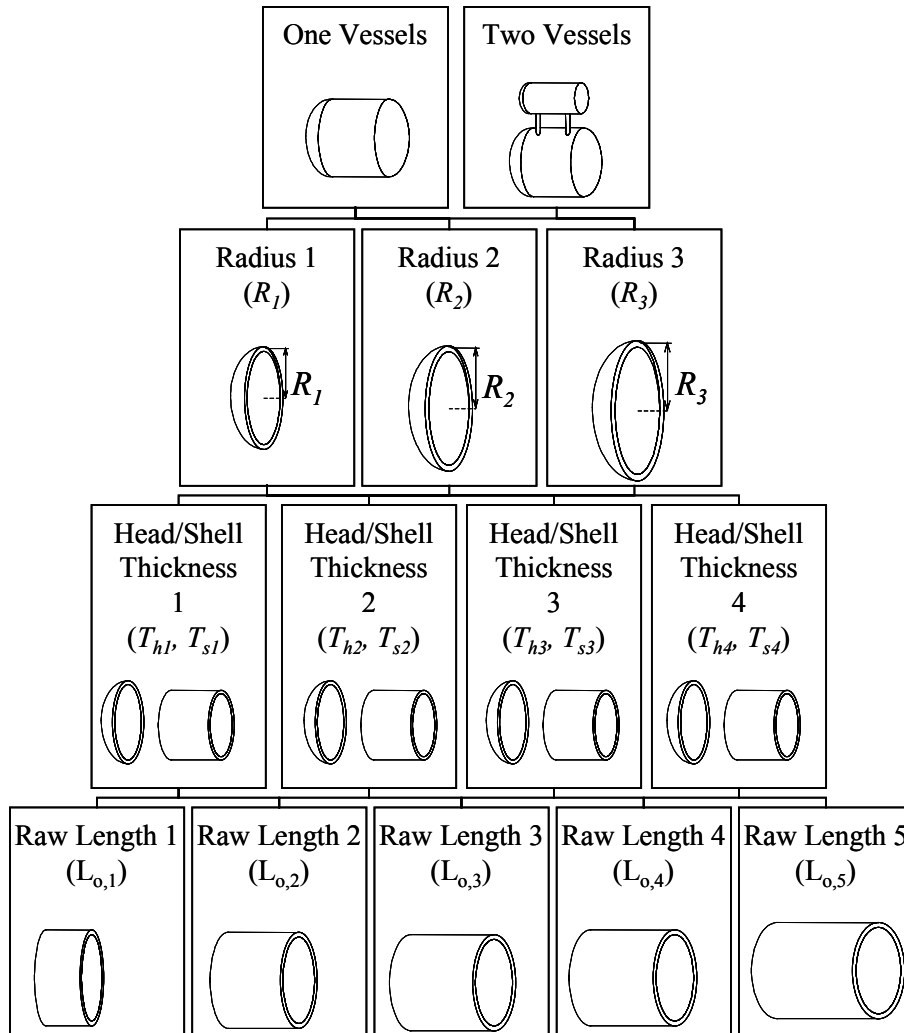
Figure 5.13: Hierarchic Organization of the Modes for Managing Product Customization for the Pressure Vessel (Williams, 2003)

Step 7: Formulate a Multi-Stage Compromise Decision Support Problem

There are two sets of objectives in this problem. Every set is comprised of two objectives that need to be satisfied – maximizing profits and minimizing variation in

profit. Since there are totally four goals, a decision has to be made that gives the best possible combination of them. The compromise DSP is used to formulate this problem.

Following the tenets of constructal theory, the determination of the range of the modes for managing variety that compose a level of the hierarchy represents one stage in a multi-stage decision. With the order of the use of the modes established in Step 6, a designer defines a cDSP for each decision stage in this step.

The manner in which the values of decision variables of the first stage $\Delta P1$ and $\Delta V1$ are obtained is explained in detail. There is a future potential expansion from 30MPa to 40MPa. In the first stage, if one considers only the current or initial market space, a particular size of the smallest element is obtained. For a particular case, assume the size is $\Delta P1$ and $\Delta V1$. Simultaneously, the potential expansion also needs to be considered. For the potential expansion, since the leveraging takes place only in the vertical direction, V1 remains the same, however; a different value of P2 is obtained that achieves the desired objectives for that extent. Depending on the probabilities of the initial and potential expansions, a compromise is developed between the sizes of the elements. From Step 2, the probability of future potential expansion is 80%. Hence, the set of objectives for the initial market space has a weight of 0.2 while the other set of objectives for the potential expansion has a weight of 0.8. Now inside each set there are two objectives. For the initial market space, the objective of maximizing profits is given a higher weight than the objective of minimizing variations, summing it up to 0.2. For the potential expansion of market space, the objectives are given equal weights of 0.4 each.
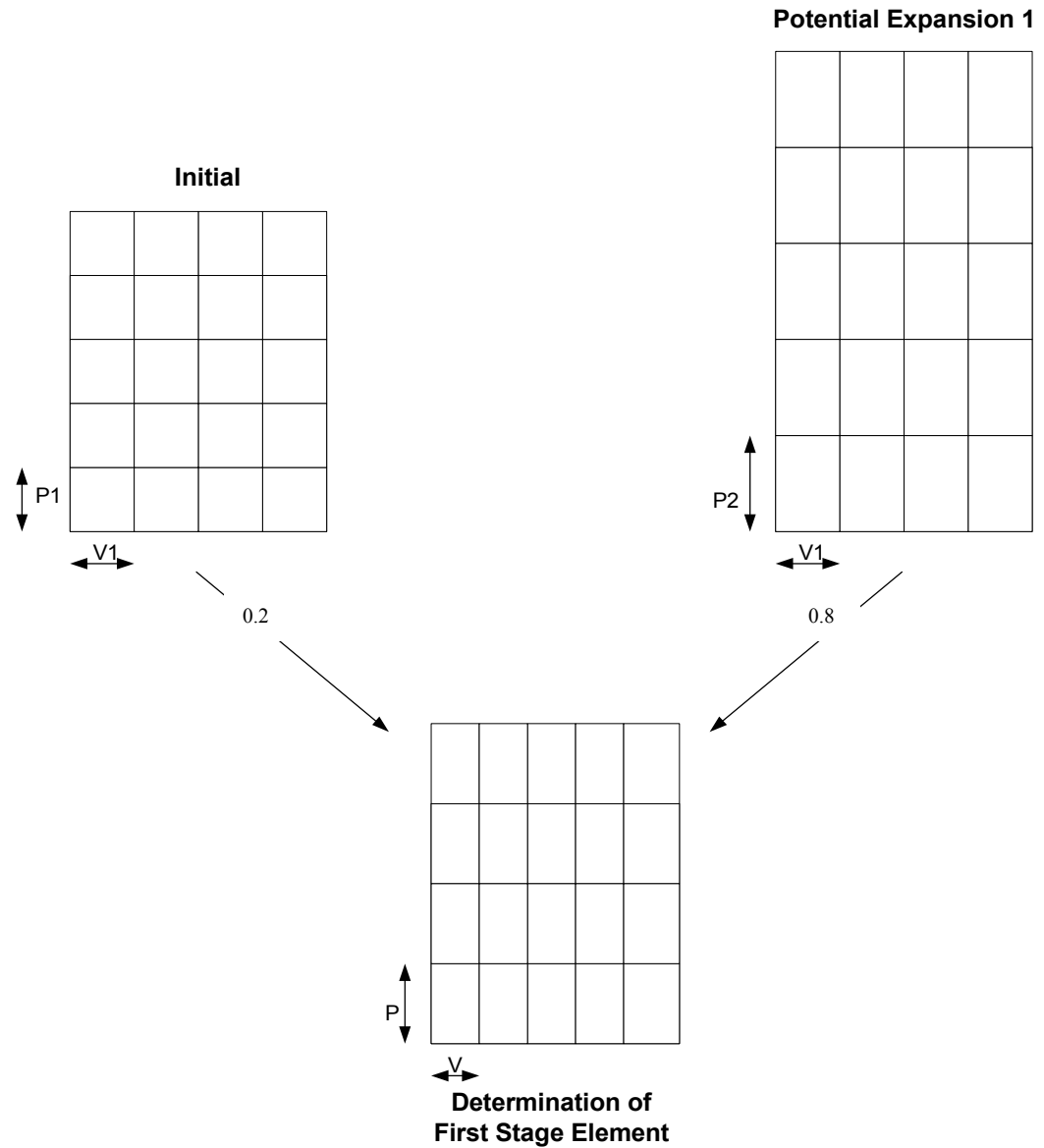
Figure 5.14: Determination of first stage element – pressure vessel

In the formulation of the compromise Decision Support Problem, the goals are measured in terms of the deviation of the objective function from the ideal value. In this case, the average profit *AvgProf* is divided by the maximum possible profit *IP* and the deviation of this ratio from the ideal value of 1 is calculated. Similarly, the minimum

possible variance *IV* is divided by the actual variance and the deviation of this ratio from the ideal value of 1 is calculated. Each set of objective function is represented by:

$$(AvgProf / IP) + d_i^- - d_i^+ = 1 \qquad\qquad [5.3]$$

$$(IV / Variance) + d_i^- - d_i^+ = 1 \qquad\qquad [5.4]$$

The deviation variables of the total four objectives are thus found. Using the weights of each objective as discussed above, along with the deviation variables forms the deviation function.

$$Z = W_{c1}\, d_{c1}^- + W_{c2}\, d_{c2}^- + W_1\, d_1^- + W_2\, d_2^- \qquad\qquad [5.5]$$

Using the formulation of the compromise DSP in this step, the size of the first stage element as P and V is determined in the next step where it is solved.

The First Stage and the First Space Element

The decision variables of the first space element are ΔV_1 and ΔP_1, the range of application of Modes V1 and P1 respectively. Three design variables are commonalized in the first space element through the application of these two modes of managing customization: the raw length, L_o, the head thickness, T_h, and the shell thickness, T_s.

Decision 1: ΔV_1, ΔP_1

Given: Current market space S = {(V, P)}

           Future extended market space S_1 = {(V_1, P_1)}

           Probability of future extended market space S_1

           Mode V1: Dimensional Customization of the Shell Length

           Mode P1: Standardization of the Head and Shell Thickness

           Mean and variance of the noise factor for each market space

Find: The value of decision variable ΔV_1, ΔP_1

           The deviation variables, d_{c1}^- and d_{c1}^+, d_{c2}^- and d_{c2}^+, d_1^- and d_1^+

Satisfy: Bounds: $0 \le \Delta V_1 \le 20$

              $0 \le \Delta P_1 \le 20$

         Constraints: $T_s \ge \left(\dfrac{P}{\sigma_y - 0.6P} \right) R$

                $T_h \ge \left(\dfrac{P}{2\sigma_y - 0.2P} \right) R$

                $d_{c1}^-, d_{c1}^+ \ge 0, \; d_{c2}^-, d_{c2}^+ \ge 0$

                $d_1^-, d_1^+ \ge 0, \; d_2^-, d_2^+ \ge 0$

                $d_{c1}^- . d_{c1}^+ = 0, \; d_{c2}^- . d_{c2}^+ = 0$

                $d_1^- . d_1^+ = 0, \; d_2^- . d_2^+ = 0$

         Goals: $(AvgProf / IP) + d_{c1}^- - d_{c1}^+ = 1$ …Contd.

...Contd. $(IV / Variance) + d_{c2}^- - d_{c2}^+ = 1$

$(AvgProf1 / IP1) + d_1^- - d_1^+ = 1$

$(IV1 / Variance1) + d_2^- - d_2^+ = 1$

Minimize: $Z = W_{c1}\, d_{c1}^- + W_{c2}\, d_{c2}^- + W_1\, d_1^- + W_2\, d_2^-$

where: *AvgProf* is calculated as in [5.1] for current market space

IP = Maximum profit of current market space

$Variance$ = Variance of current market space

IV = Minimum variance of current market space

AvgProf1 is calculated as in [5.1] for future market space S_1

$IP1$ = Maximum profit of future market space S_1

Variance1 = Variance of future market space S_1

$IV1$ = Minimum variance of future market space S_1

Figure 5.15: Formulation of the Multistage Compromise Decision Support Problem for the Pressure Vessel; First Stage Element

The Second Stage and the Second Space Element

The decision variable of the second space element is ΔV_2, the range of application of Mode V2. Only one design variable is commonalized in the second space element: the radius, R.

248

Decision 2: ΔP_2

Given: Current market space $S = \{(V, P)\}$

Future extended market space $S_1 = \{(V_1, P_1)\}$

Probability of future extended market space S_1

The value of ΔV_1, and ΔP_1 from the first stage

Mode P2: Commonalization of the Radius

Mean and variance of the noise factor for each market space

Find: The value of decision variable ΔP_2

The deviation variables, d_{c1}^- and d_{c1}^+, d_{c2}^- and d_{c2}^+, d_1^- and d_1^+, d_2^- and d_2^+

Satisfy: Bounds: $0 \leq P2 \leq 20$

Constraints: $\Delta P_1 \leq \Delta P_2 \leq 20$

$$T_s \geq \left(\frac{P}{\sigma_y - 0.6P} \right) R$$

$$T_h \geq \left(\frac{P}{2\sigma_y - 0.2P} \right) R$$

$$d_{c1}^-, d_{c1}^+ \geq 0$$

$$d_{c2}^-, d_{c2}^+ \geq 0$$

$$d_1^-, d_1^+ \geq 0$$

$$d_2^-, d_2^+ \geq 0$$

$$d_{c1}^- . d_{c1}^+ = 0 \qquad\qquad \text{…Contd.}$$

...Contd.
$$d_{c2}^- \cdot d_{c2}^+ = 0$$

$$d_1^- \cdot d_1^+ = 0$$

$$d_2^- \cdot d_2^+ = 0$$

Goals:
$$(AvgProf / IP) + d_{c1}^- - d_{c1}^+ = 1$$

$$(IV / Variance) + d_{c2}^- - d_{c2}^+ = 1$$

$$(AvgProf1 / IP1) + d_1^- - d_1^+ = 1$$

$$(IV1 / Variance1) + d_2^- - d_2^+ = 1$$

Minimize: $Z = W_{c1}\, d_{c1}^- + W_{c2}\, d_{c2}^- + W_1\, d_1^- + W_2\, d_2^-$

where: *AvgProf* is calculated as in [5.1] for current market space

IP = Maximum profit of current market space

Variance = Variance of current market space

IV = Minimum variance of current market space

AvgProf1 is calculated as in [5.1] for future market space S_1

IP1 = Maximum profit of future market space S_1

Variance1 = Variance of future market space S_1

IV1 = Minimum variance of future market space S_1

Figure 5.16: Formulation of the Multistage Compromise Decision Support Problem for the Pressure Vessel; Second Stage Element

The Third Stage and the Third Space Element

The third and final space element is defined by the range of application of Mode V3: Modular Combination of Vessels. The decision associated with this mode is the selection of "cut-off" point – a point where vessels with larger volumes are composed of two smaller vessels below the cut-off point. As such, the space element S_3 is composed

of a number of spaces S_2 from the minimum volume of the space, V_{min}, to a cut-off point,

V'. The cDSP for this stage is presented in Figure 5.17.

Decision 3: ΔV_2

Given: Current market space S = {(V, P)}

        Future extended market space S_1 = {(V_1, P_1)}

        Probability of future extended market space S_1

        The value of ΔV_1, ΔP_1, ΔP_2 from the first and second stages

        Mode V2: Modular Combination of Vessels

        Mean and variance of the noise factor

Find: The value of decision variable ΔV_2

        The deviation variables, d_{c1}^- and d_{c1}^+, d_{c2}^- and d_{c2}^+, d_1^- and d_1^+

Satisfy: Bounds: $0 \leq V2 \leq 20$

      Constraints: $\Delta V_1 \leq \Delta V_2 \leq 20$

$$T_s \geq \left(\frac{P}{\sigma_y - 0.6P} \right) R$$

$$T_h \geq \left(\frac{P}{2\sigma_y - 0.2P} \right) R$$

$$d_{c1}^-, d_{c1}^+ \geq 0$$

$$d_{c2}^-, d_{c2}^+ \geq 0$$

$$d_1^-, d_1^+ \geq 0 \qquad\qquad \text{…Contd.}$$

…Contd. $\qquad$ $d_2^-, d_2^+ \geq 0$

$$d_{c1}^- . d_{c1}^+ = 0$$

$$d_{c2}^- . d_{c2}^+ = 0$$

$$d_1^- . d_1^+ = 0$$

$$d_2^- . d_2^+ = 0$$

Goals: $\qquad$ $(AvgProf / IP) + d_{c1}^- - d_{c1}^+ = 1$

$$(IV / Variance) + d_{c2}^- - d_{c2}^+ = 1$$

$$(AvgProf1 / IP1) + d_1^- - d_1^+ = 1$$

$$(IV1 / Variance1) + d_2^- - d_2^+ = 1$$

Minimize: $Z = W_{c1} d_{c1}^- + W_{c2} d_{c2}^- + W_1 d_1^- + W_2 d_2^-$

where: *AvgProf* is calculated as in [5.1] for current market space

IP = Maximum profit of current market space

Variance = Variance of current market space

IV = Minimum variance of current market space

AvgProf1 is calculated as in [5.1] for future market space S_1

$IP1$ = Maximum profit of future market space S_1

Variance1 = Variance of future market space S_1

$IV1$ = Minimum variance of future market space S_1

Figure 5.17: Formulation of the Multistage Compromise Decision Support Problem for the Pressure Vessel; Third Stage Element

With the multi-stage cDSP formulated, we move on to the final step of the augmented PPCTM: its solution.

Step 8: Solve the Multi-Stage Compromise Decision Support Problem

The final step of this method is the formulation of an appropriate solution technique. Any appropriate solution algorithm can be used; its primary goal is the determination of the extent of application of the modes that provides the least deviation function, from which the commonality of design parameters of the product platform is determined. Our solution technique, an exhaustive search, is illustrated in Figure 5.18 (modified from Williams, 2003).

This solution method involves iterating through values of the modes of managing variety (ΔV_1, ΔV_2, ΔV_3, and ΔP_1), establishing the dimensions of the sub-spaces, commonalizing the design parameters (L_o, T_s, T_h, and R) across each sub-space, evaluating the objective functions, and calculating and comparing the resulting overall deviation function for each iteration. Specifically:

- A particular combination of sub-space element sizes (ΔV_1, ΔV_2, ΔV_3, and ΔP_1) is taken as input **(A)**. The input also includes demand scenario **(B)** (present and future) (Step 1), current geometric space (market space) (Step 1) and future potential expansion of the market with its probability **(C)** (Step 2) and the objective functions **(D)** (maximize profit, minimize variance) (Step 4).

- From the sizes of the sub-space elements **(E)**, the bounds of each sub-space are established **(F)**. The values of parameters to be commonalized are determined from the bounds of the sub-space **(G)**. This involves calculation of the parameters from their respective modes **(I, J, K)** (as shown in Section 4.1.2).

253

- These parameters are then commonalized across the respective sub-spaces **(H)**. The objective function is determined for every variant considering the parameter values across each sub space **(L)**.

- For calculation of the variance in profit, the first step involves calculation of cost from the parameter values **(M)** and assessing the selling price **(N)** and demand for each variant. Then variance is provided to demand in the form of noise. The profit is then calculated while keeping other parameters constant. From the profit obtained from different values of demand, the variance in profit is calculated **(O)**. Specific to this example problem, ±20% noise is provided to the values of demand in four discrete levels (i.e., Demand (Noise) = {0.8D, 0.9D, 1.1D, 1.2D}, where D is the mean demand).

- The total profit (Equation 5.1) and variance in profit (Equation 5.2) is calculated for that particular combination of sub-space element taken as input **(P)**. The objectives are found for the initial as well as potential expansion of the market.

- The deviation variables of each of the objectives are obtained **(Q, R, S, T)**. Depending on the probability of the future expansion, the weights for each of the objective of cDSP are decided. The deviation variables and the weights of profit and variance for the initial and potential market spaces help to obtain the deviation function as the output of this algorithm **(U)**.

- The value of the deviation function is returned from the analysis and is compared with previous output. After each feasible combination of ranges of the modes for managing variety is explored, the input that provides the smallest deviation from the goal is selected.
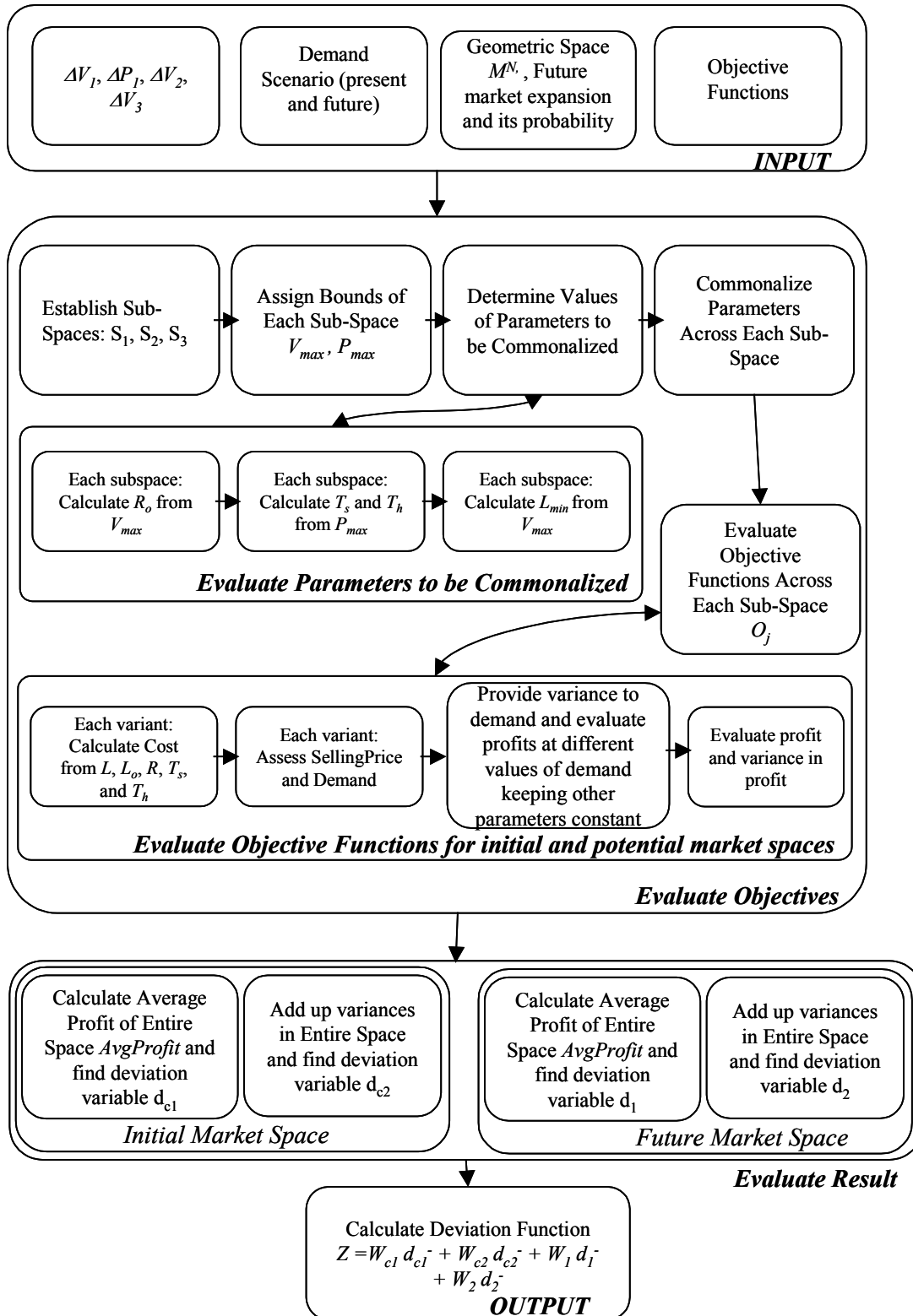
Figure 5.18: Solution Algorithm for Minimizing Variance in the Pressure Vessel example

5.3 RESULTS AND ANALYSIS IN THE PRESSURE VESSEL PROBLEM
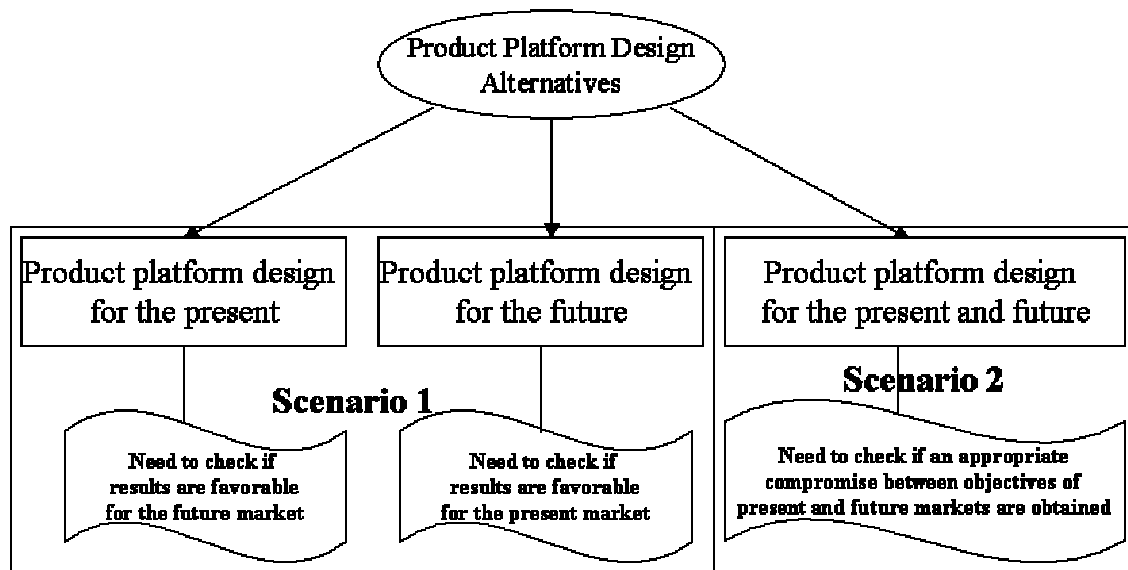
Figure 5.19: Product Platform Design Alternatives

There are two scenarios that are analyzed. The first scenario considers two different cases. In one case, the platform is designed considering only the present market, while in the other case only the future market is considered. Basically, the first scenario deals with product platform design where one can only consider a particular extent of the market space. The results of this scenario act as a comparison to the second scenario where the platform design is done considering both present as well as future markets simultaneously. In this second scenario, the product platform is designed using the method proposed in this paper. This scenario helps to show how a compromise between

the objectives of the present and future markets is made to design a strategic product platform.

5.3.1 Scenario 1 – Design considering Present Market OR Future Market

In the first scenario, we consider a manufacturer of pressure vessels having a market defined by volume in the range of 10-30 m^3 and pressure in the range of 10-30 MPa. This is the current market situation for the manufacturer. So the manufacturer designs a product platform using the PPCTM in this marketplace and obtains the results as shown in Table 5.1.

In Figure 5.20, the values of ΔV_1, ΔV_2, ΔV_3, and ΔP_1 define a product family; the corresponding value of the deviation function Z of that particular family is also noted down. The results inform the manufacturer that the best configuration of the modes of managing variety which closely matches his/her preferences is to commonalize the raw length for every 0.5 m^3 of volume, commonalize the head and shell thickness for every 5 MPa of pressure, commonalize the radius for every 2.5 m^3, and to not modularly combine the vessels. It is seen that the market space is divided into subspaces and the size of the subspaces depend on the values of ΔV_1, ΔV_2, ΔV_3, and ΔP_1.

ΔV3 (m^3)	20
ΔV2 (m^3)	2.5
ΔP1 (MPa)	0.5
ΔV1 (m^3)	0.5
Deviation Z	0.4389
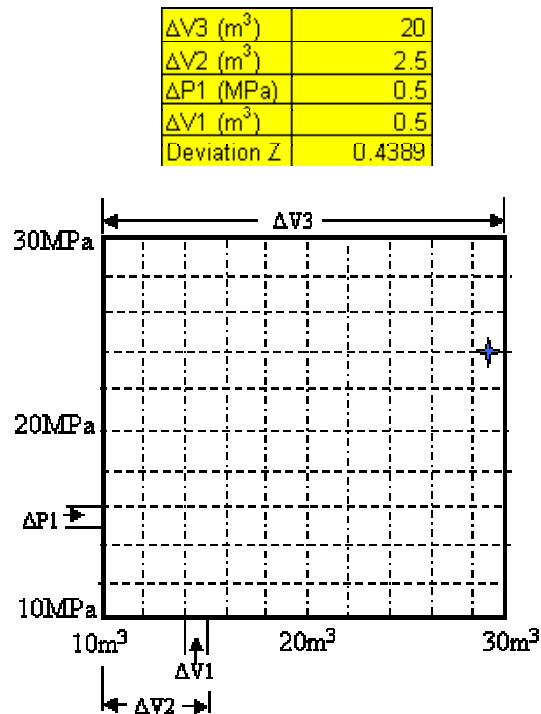
Figure 5.20: ΔV1, ΔP1, ΔV2 and ΔV3 – Scenario 1 Product Platform Results
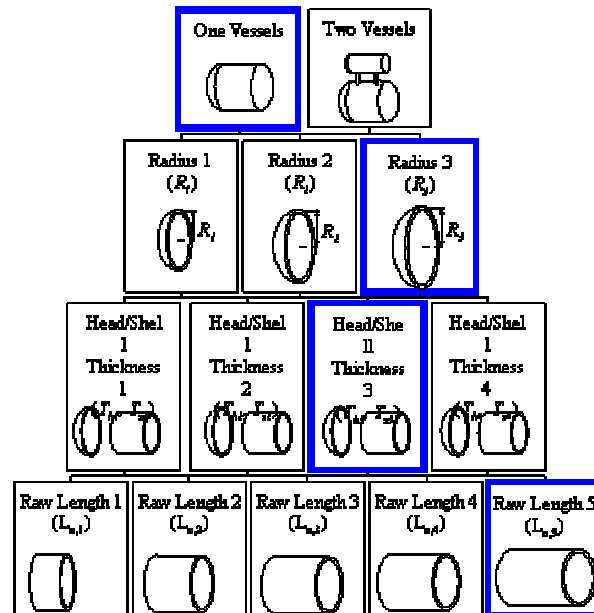
Figure 5.21: Hierarchical Arrangement of the Product Platform

Table 5.1: Section of the Scenario 1 Product Platform Specifications

V (m³)	P (MPa)	R (m) (ΔV2=2.5m	Ts (mm) (ΔP1=2MPa	Th (mm) (ΔP1=2MP	Lo (m) (ΔV1=0.5
29.5	10	1.086	10.140	5.046	6.514
	10.5		10.650	5.299	
	11		11.160	5.552	
	11.5		11.671	5.804	
	12		12.182	6.057	
	12.5		12.693	6.310	
	13		13.204	6.562	
	13.5		13.716	6.815	
	14		14.228	7.068	
	14.5		14.740	7.320	
	15		15.253	7.573	
	15.5		15.766	7.826	
	16		16.279	8.079	
	16.5		16.792	8.332	
	17		17.306	8.585	
	17.5		17.820	8.837	
	18		18.334	9.090	
	18.5		18.849	9.343	
	19		19.364	9.596	
	19.5		19.879	9.849	
	20		20.394	10.102	
	20.5		20.910	10.355	
	21		21.426	10.608	
	21.5		21.942	10.862	
	22		22.459	11.115	
	22.5		22.976	11.368	
	23		23.493	11.621	
	23.5		24.011	11.874	
	24		24.529	12.127	
	24.5		25.047	12.381	
	25		25.565	12.634	
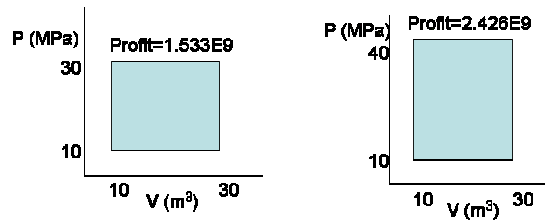
Figure 5.22: Profits when designing only for the Initial Market Space

259

Consider a customer having volume and pressure requirements of a pressure vessel as 29.5m^3 and 24MPa respectively (denoted by star in Figure 5.20). Using the different modes of managing variety at each hierarchical level (Figure 5.21), one obtains the product platform specifications involving R, T_s, T_h and L from the particular values of ΔV_1, ΔV_2, ΔV_3, and ΔP_1. A section of the design specification of the product family of this scenario is presented in Table 5.1. The columns in Table 5.1 represent the extent of commonalization for each design variable. It is observed that the figures in bold in Table 5.1 satisfy the requirements of the customer. The detailed specifications of the product platform of this scenario are provided in Appendix B.

The total profit obtained by the manufacturer in this current marketplace is given in the LHS of Figure 5.22. If the same values of ΔV_1, ΔV_2, ΔV_3, and ΔP_1 are used in the extension of the market space in the pressure range 30-40MPa, the total profit obtained in the entire market space is given in the RHS of Figure 5.22.

The problem is better analyzed by considering Figure 5.23 where the product platform is designed for the future market and what if the same results are used in the present market is analyzed. The product platform for the new market of pressure range 10-40MPa is designed and the values of ΔV_1, ΔV_2, ΔV_3, and ΔP_1 are obtained as shown in Figure 5.23. Considering the future market, the profit obtained by this new product platform in the market space having pressure range 10-40MPa (RHS of Figure 5.23) is obviously greater than the profit obtained by extension of the platform designed for pressure range 10-30MPa (RHS of Figure 5.22). The reason is that the new platform is designed for the bounds of this new market place, while the previous one was just an extension of the existing platform designed for the present market place. Considering the

present market, the profit obtained by the product platform designed for the present

market having pressure range 10-30MPa (LHS of Figure 5.22) is greater than the profit

obtained by the product platform designed for the extended market space having pressure

range 10-40MPa (LHS of Figure 5.23).

$\Delta V3$ (m^3)	10
$\Delta V2$ (m^3)	5
$\Delta P1$ (MPa)	0.5
$\Delta V1$ (m^3)	5
Deviation f	0.5487

A **B** **C**
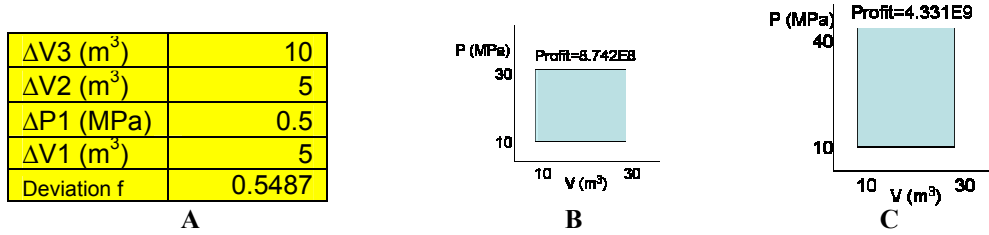
Figure 5.23: Results when designing only for potential market space

5.3.2 Scenario 2 – Design considering Present AND Future Markets

In this scenario, while designing the initial product platform for the current market

space, the manufacturer considers the possible future market condition. Considering the

probability of different market situations in future, the manufacturer predicts the most

probable market extension of the future. Here, let us assume that the demand will be

higher in the range of 30-40MPa. So in this case, the manufacturer strategically designs

the initial product platform using the augmented method discussed in Section 3.3. This

consideration helps the manufacturer not to fall into problems discussed in Scenario 1.

Table 5.2: Product Platforms Generated during Iterations

| | $\Delta V3$ (m^3) | $\Delta V2$ (m^3) | $\Delta P1$ (MPa) | $\Delta V1$ (m^3) | Current Market | | Potential Market | | Deviation function |
					Profit ($)	Variance (^{2})	Profit ($)	Variance (^{2})	
1	10	0.5	0.5	0.5	1.148E+09	3.117E+16	2.037E+09	9.381E+16	0.743
2	20	1	0.5	1	1.522E+09	4.515E+16	2.220E+09	9.780E+16	0.837
3	10	5	0.5	2.5	1.103E+09	2.949E+16	2.526E+09	1.067E+17	0.700
4	10	1	1	0.5	1.117E+09	2.995E+16	2.823E+09	1.214E+17	0.676
5	10	2	5	1	8.789E+08	2.161E+16	4.183E+09	2.494E+17	0.561
6	20	10	5	1	1.298E+09	3.305E+16	4.223E+09	2.600E+17	0.769
7	20	10	10	10	8.933E+08	1.768E+16	2.343E+09	1.352E+17	0.845
8	20	0.5	10	0.5	1.067E+09	2.372E+16	3.428E+09	1.900E+17	0.604
9	20	5	0.5	5	1.123E+09	3.000E+16	4.327E+09	2.709E+17	0.547

In Table 5.2, some of the possible different product families obtained during the iterations is noted down. As explained in Scenario 1, it is seen in Table 5.2 that each of the rows represents a product family. The values of the two objectives of profit and variance in the case of current market and potential market for each product family are calculated. The compromise DSP helps to achieve a compromise between these values according to the weights assigned to each of the objectives. The lower the value of the deviation function, the product family is closer to the ideal values of the objectives. As discussed in Step 7, the weight given to the future potential expansion is more than the current market situation and hence better results for the potential market are preferred. Moreover, between the objectives of profit and variance, higher profit is preferred. For better understanding, further results are analyzed considering the profits of each situation. It is seen that in some of the product families in Table 5.2 (Example 8), one is able to get high profit for the potential market situation however since the current market profit values are low, the deviation function is high. In some of the cases (Example 2), the current market profit is high however since the potential market profit is not so high as well as the weight for the potential market being high the deviation function is high. In cases where both the profits are considerably high, the variance values also come into consideration. The Example 9 in Table 5.2 has the lowest deviation function and thus makes the best possible compromise between the different values. Now this product family is explained in more detail in Figure 5.24 and compared with the Scenario 1 to see its effectiveness.

ΔV3 (m³)	15
ΔV2 (m³)	5
ΔP1 (MPa)	0.5
ΔV1 (m³)	5
Deviation Z	0.5472
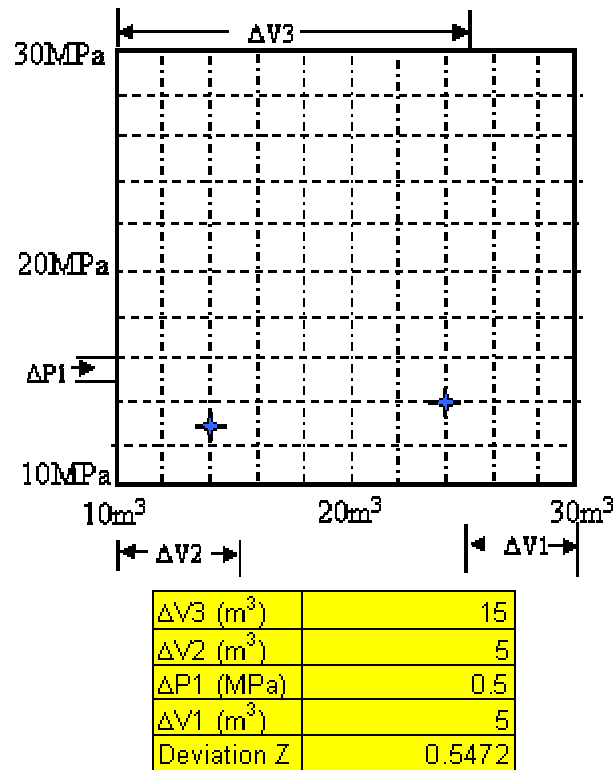
Figure 5.24: ΔV1, ΔP1, ΔV2 and ΔV3 – Scenario 2 Product Platform Results

Table 5.3: Section of the Scenario 2 Product Platform Specifications (14m³ and 12.5MPa)

V (m³)	P (MPa)	R (m) (ΔV2=5m³)	Ts (mm) (ΔP1=0.5MPa)	Th (mm) (ΔP1=0.5MPa)	Lo (m) (ΔV1=5m³)
15	10	0.783	7.311	3.638	6.744
	10.5		7.679	3.821	
	11		8.047	4.003	
	11.5		8.415	4.185	
	12		8.783	4.367	
	12.5		9.151	4.549	
	13		9.520	4.731	
	13.5		9.889	4.914	
	14		10.258	5.096	
	14.5		10.628	5.278	
	15		10.997	5.460	

Figure 5.25: Hierarchical Arrangement of the Product Platform (14m^3 and 12.5MPa)

Table 5.4: Section of the Scenario 2 Product Platform Specifications (24m^3 and 14MPa)

V (m^3)	P (MPa)	R (m) (ΔV2=5m^3)	Ts (mm) (ΔP1=0.5MPa)	Th (mm) (ΔP1=0.5MPa)	Lo (m) (ΔV1=5m^3)
20	10	0.898	8.385	4.173	6.697
	10.5		8.806	4.382	
	11		9.228	4.591	
	11.5		9.650	4.799	
	12		10.073	5.008	
	12.5		10.496	5.217	
	13		10.918	5.426	
	13.5		11.342	5.635	
	14		11.765	5.844	
	14.5		12.189	6.053	
	15		12.612	6.262	

264

Figure 5.26: Hierarchical Arrangement of the Product Platform (24m^3 and 14MPa)
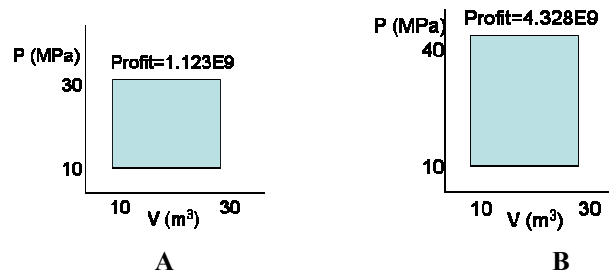
Figure 5.27: Profits when Designing Considering the Initial and Potential Market Space (Compromise)

The results inform the manufacturer that the best configuration of the modes of managing variety which closely matches his/her preferences is to commonalize the raw length for every 5 m^3 of volume, commonalize the head and shell thickness for every 0.5 MPa of pressure, commonalize the radius for every 5 m^3, and to modularly combine the vessels if the volume is greater than 15 m^3.

Here, two customer examples are shown.

- Consider a customer having volume and pressure requirements of a pressure vessel as 14m^3 and 12.5MPa respectively (denoted by left star in Figure 5.24). Using the different modes of managing variety at each hierarchical level (Figure 5.25), one obtains the product platform specifications involving R, T_s, T_h and L from the particular values of ΔV_1, ΔV_2, ΔV_3, and ΔP_1. A section of the design specification of the product family of this scenario is presented in Table 5.3. It is observed that the figures in bold in Table 5.3 satisfy the requirements of the customer. The vessel of the product family that will satisfy the customer requirements is the one having volume as 15m^3 and pressure as 12.5MPa. The radius of this vessel is 0.783m, thickness of shell is 9.151mm, thickness of head is 4.549mm and raw length of sheet is 6.744m

- Consider another customer having volume and pressure requirements of a pressure vessel as 24m^3 and 14MPa respectively (denoted by right star in Figure 5.24). A section of the design specification of the product family of this scenario is presented in Table 5.4. It is observed that the figures in bold in Table 5.4 satisfy the requirements of the customer. The vessel of the product family that will satisfy the customer requirements is the one having volume as 25m^3 and pressure as 14MPa. The radius of this vessel is 0.998m, thickness of shell is 13.069mm, thickness of head is 6.492mm and raw length of sheet is 6.668m.

It is important to note in both the examples the customer is satisfied by having the pressure and volume of a pressure vessel higher than his/her requirements. The total profit obtained by the manufacturer in this current marketplace is given in the LHS of Figure 5.27. If the same values of ΔV_1, ΔV_2, ΔV_3, and ΔP_1 are used in the extension of the

market space in the pressure range 30-40MPa, the total profit obtained in the entire market space is given in the RHS of Figure 5.27.

After a certain time interval, when the market needs to be extended to have pressure in the range of 10-40MPa, the existing product platform is extended to capture this new market. There will not be any restructuring costs to move into the market space since it is continuing with the same product platform (Option 1 – Section 1.2, 4.1). It is seen by comparing RHS of Figure 5.27 with the RHS of Figure 5.22 the profit has been greater than the profit that would have been obtained by extension of the product portfolio designed for the present market without consideration of the future extension (Option 2 – Section 1.2, 4.1). However, since a compromise is achieved between the objectives of the two product platforms, it is seen that the profit obtained in the initial market extent for scenario 2 (LHS of Figure 5.27) is less than that of scenario 1 (LHS of Figure 5.22). It must also be noted that the profit obtained in the initial market extent for scenario 2 (LHS of Figure 5.27) is more than the profit obtained in the present market using the product platform designed for the extended market space (LHS of Figure 5.23).

Table 5.5: Product Platform Specifications for Scenario 2

V (m³)	P (MPa)	R (m) (ΔV2=5m³)	Ts (mm) (ΔP1=0.5 MPa)	Th (mm) (ΔP1=0.5 MPa)	Lo (m) (ΔV1=5m³)
15	10	0.783	7.311	3.638	6.744
	10.5		7.679	3.821	
	11		8.047	4.003	
	11.5		8.415	4.185	
	12		8.783	4.367	
	12.5		9.151	4.549	
	13		9.520	4.731	
	13.5		9.889	4.914	
	14		10.258	5.096	
	14.5		10.628	5.278	

Table 5.5 Contd..

	15		10.997	5.460	
	15.5		11.367	3.638	
	16		11.737	3.821	
	16.5		12.107	4.003	
	17		12.478	4.185	
	17.5		12.848	4.367	
	18		13.219	4.549	
	18.5		13.590	4.731	
	19		13.961	4.914	
	19.5		14.333	5.096	
	20		14.704	7.284	
	20.5		15.076	5.460	
	21		15.448	5.643	
	21.5		15.820	5.825	
	22		16.193	6.007	
	22.5		16.566	6.189	
	23		16.938	6.372	
	23.5		17.312	6.554	
	24		17.685	6.737	
	24.5		18.058	6.919	
	25		18.432	9.109	
	25.5		18.806	7.284	
	26		19.180	7.466	
	26.5		19.555	7.649	
	27		19.929	7.831	
	27.5		20.304	8.014	
	28		20.679	8.196	
	28.5		21.054	8.379	
	29		21.430	10.570	
	29.5		21.805	8.744	
	30		22.181	10.936	
20	10	0.898	8.385	4.173	6.697
	10.5		8.806	4.382	
	11		9.228	4.591	
	11.5		9.650	4.799	
	12		10.073	5.008	
	12.5		10.496	5.217	
	13		10.918	5.426	
	13.5		11.342	5.635	
	14		11.765	5.844	
	14.5		12.189	6.053	
	15		12.612	6.262	
	15.5		13.036	6.471	
	16		13.461	6.680	
	16.5		13.885	6.889	
	17		14.310	7.098	
	17.5		14.735	7.308	

Table 5.5 Contd..

	18		15.160	7.517	
	18.5		15.586	7.726	
	19		16.012	7.935	
	19.5		16.438	8.144	
	20		16.864	8.353	
	20.5		17.290	8.563	
	21		17.717	8.772	
	21.5		18.144	8.981	
	22		18.571	9.191	
	22.5		18.999	9.400	
	23		19.426	9.609	
	23.5		19.854	9.819	
	24		20.282	10.028	
	24.5		20.711	10.237	
	25		21.139	10.447	
	25.5		21.568	10.656	
	26		21.997	10.866	
	26.5		22.427	11.075	
	27		22.856	11.285	
	27.5		23.286	11.494	
	28		23.716	11.704	
	28.5		24.147	11.913	
	29		24.577	12.123	
	29.5		25.008	12.332	
	30		25.439	12.542	
25	10	0.998	9.318	4.638	6.659
	10.5		9.787	4.870	
	11		10.256	5.102	
	11.5		10.725	5.334	
	12		11.195	5.566	
	12.5		11.664	5.798	
	13		12.134	6.030	
	13.5		12.605	6.263	
	14		13.075	6.495	
	14.5		13.546	6.727	
	15		14.017	6.960	
	15.5		14.488	7.192	
	16		14.960	7.424	
	16.5		15.432	7.657	
	17		15.904	7.889	
	17.5		16.376	8.121	
	18		16.849	8.354	
	18.5		17.322	8.586	
	19		17.795	8.819	
	19.5		18.268	9.051	
	20		18.742	9.284	
	20.5		19.216	9.516	

Table 5.5 Contd..

	21		19.690	9.749	
	21.5		20.164	9.981	
	22		20.639	10.214	
	22.5		21.114	10.447	
	23		21.590	10.679	
	23.5		22.065	10.912	
	24		22.541	11.145	
	24.5		23.017	11.377	
	25		23.493	11.610	
	25.5		23.970	11.843	
	26		24.447	12.076	
	26.5		24.924	12.308	
	27		25.402	12.541	
	27.5		25.879	12.774	
	28		26.357	13.007	
	28.5		26.836	13.240	
	29		27.314	13.473	
	29.5		27.793	13.706	
	30		28.272	13.939	
30	10	1.087	7.313	3.639	14.523
	10.5		7.681	3.822	
	11		8.049	4.004	
	11.5		8.417	4.186	
	12		8.785	4.368	
	12.5		9.154	4.550	
	13		9.523	4.733	
	13.5		9.892	4.915	
	14		10.261	5.097	
	14.5		10.630	5.279	
	15		11.000	5.462	
	15.5		14.488	7.192	
	16		14.960	7.424	
	16.5		15.432	7.657	
	17		15.904	7.889	
	17.5		16.376	8.121	
	18		16.849	8.354	
	18.5		17.322	8.586	
	19		17.795	8.819	
	19.5		18.268	9.051	
	20		18.742	9.284	
	20.5		19.216	9.516	
	21		19.690	9.749	
	21.5		20.164	9.981	
	22		20.639	10.214	
	22.5		21.114	10.447	
	23		21.590	10.679	
	23.5		22.065	10.912	
	24		22.541	11.145	

Table 5.5 Contd..

	24.5		23.017	11.377	
	25		23.493	11.610	
	25.5		23.970	11.843	
	26		24.447	12.076	
	26.5		24.924	12.308	
	27		25.402	12.541	
	27.5		25.879	12.774	
	28		26.357	13.007	
	28.5		26.836	13.240	
	29		27.314	13.473	
	29.5		27.793	13.706	
	30		28.272	13.939	

Thus, it is observed from the results of Scenario 2 that a compromise is achieved between the objectives of initial and potential market situations. The product platform designed will have desirable performance not only for the starting period but also later on when market extension takes place. Thus one saves on restructuring costs and extends his/her market according to requirements.

Additional Scenarios

In Scenario 2, the probability for the future extension considered is 80%, while the probability for the manufacturer to continue in the same market extent is 20% (as stated in Step 2 of PPCTM-RCM). In this part, some additional scenarios are run for different market probabilities of extension and towards the end all the scenarios are analyzed together.

Scenario 3 – Probability of current extent 80% and future extent 20%

In this scenario, manufacturer assigns a probability of 80% for the market to continue in the same current extent. The probability to extend into the neighboring

market space is 20%. In this case the product platform is designed and the results are seen in Figure 5.28.

ΔV3 (m³)	20
ΔV2 (m³)	1
ΔP1 (MPa)	2.5
ΔV1 (m³)	1
Deviation Z	0.4842

A B C
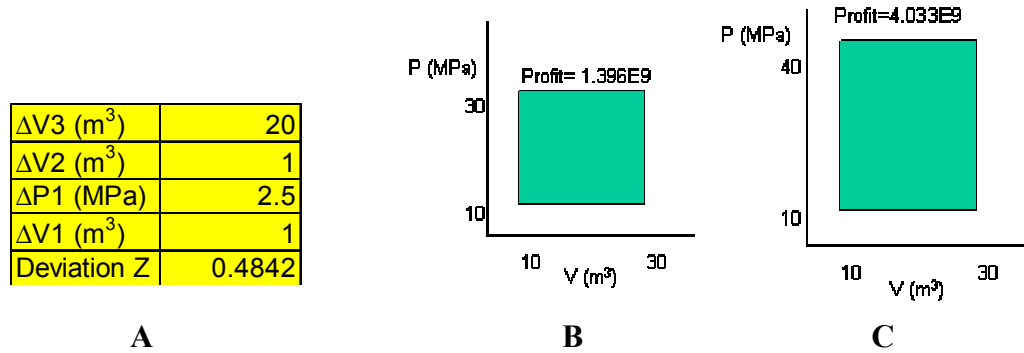
Figure 5.28: ΔV1, ΔP1, ΔV2 and ΔV3 – Scenario 3 Product Platform Results

It is seen in Figure 5.28 that the current extent profit ($1.396E9) is more than the profit for the current extent ($1.123E9) obtained in Scenario 2 (Figure 5.27) and the profit for the future extent ($4.033E9) is less than the corresponding profit ($4.328E9) in Scenario 2. This is the case because now the probability of the current extent has been increased from 20% in Scenario 2 to 80% in this scenario and probability of the future extent is decreased from 80% in Scenario 2 to 20% in this scenario. One can note that the profit for the current extent obtained in this scenario is closer to the maximum profit ($1.533E9) that can be obtained for the current extent in Scenario1 – Product Platform Design for the Present (Figure 5.22). Also, the profit for the future extent obtained in this scenario is further away from the maximum profit ($4.33E9) that can be obtained for the future extent in Scenario 1 – Product Platform Design for the Future (Figure 5.23).

Scenario 4 – Probability of current extent 40% and future extent 60%

In this scenario, manufacturer assigns a probability of 40% for the market to continue in the same current extent. The probability to extend into the neighboring

market space is 60%. In this case the product platform is designed and the results are seen in Figure 5.29.

ΔV3 (m³)	20
ΔV2 (m³)	2.5
ΔP1 (MPa)	5
ΔV1 (m³)	0.5
Deviation Z	0.5263

A   B   C
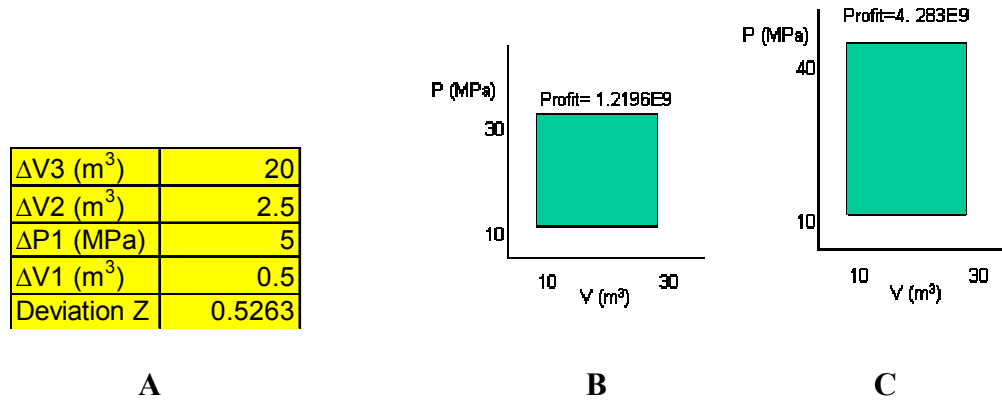
Figure 5.29: Δ V1, ΔP1, ΔV2 and ΔV3 – Scenario 4 Product Platform Results

It is seen in Figure 5.29 that the current extent profit ($1.2196E9) is more than the profit for the current extent ($1.123E9) obtained in Scenario 2 (Figure 5.27) and less than corresponding profit ($1.396E9) in Scenario 3 (Figure 5.28) and the profit for the future extent ($4.283E9) is less than the corresponding profit in Scenario 2 ($4.328E9) and more than the corresponding profit ($4.033E9) in Scenario 3. This is the case because now the probability of the current extent and future extent are in between that in Scenario 2 and Scenario 3.

Economic Gains over a Five Year Period

Based on the different scenarios, one can now analyze the results over a five year period. Here, what is done is two manufacturers A and B placed in different situations are imagined. In each of the situation, the profits that the manufacturer would have gained after 5 years if he/she uses PPCTM-RCM (with different probabilities of market extension) and if he/she did not use the method are considered. This helps to understand

how much exactly the manufacturers would gain after 5 years based on their market prediction.

First, consider a manufacturer A designing a pressure vessel platform who predicts the current extent and future extent of the market having a probability of 20% and 80% respectively. We assume that the manufacturer is correct in his prediction and the higher pressure market is profitable after a year and he/she extends his platform in the neighboring market. He/she gains profits in the extended market space for the next four years.

The manufacturer A has a number of options for designing the product platform in the current market space at the start of the five year period. As discussed previously in this section, he/she can use the non-augmented PPCTM and design the product platform as shown in the two cases of Scenario 1 or he/she can use the augmented PPCTM-RCM and design the product platform. All the possible cases are compared by using the results obtained in different scenarios considered till now (Table 5.6). The final profit obtained by the manufacturer after the five year period is calculated for the different scenarios.

Table 5.6: Total Profits in Different Probability Scenarios after 5 Years for Manufacturer A

	Scenario 1 - Present	Scenario 2	Scenario 3	Scenario 1 - Future
	Current 100% Future 0%	Current 20% Future 80%	Current 80% Future 20%	Current 0% Future 100%
Year 1	1.533E+09	1.123E+09	1.396E+09	8.742E+08
Year 2	2.425E+09	4.328E+09	4.033E+09	4.330E+09
Year 3	2.425E+09	4.328E+09	4.033E+09	4.330E+09
Year 4	2.425E+09	4.328E+09	4.033E+09	4.330E+09
Year 5	2.425E+09	4.328E+09	4.033E+09	4.330E+09
Total Profit After 5 Years	1.123E+10	1.843E+10	1.753E+10	1.819E+10
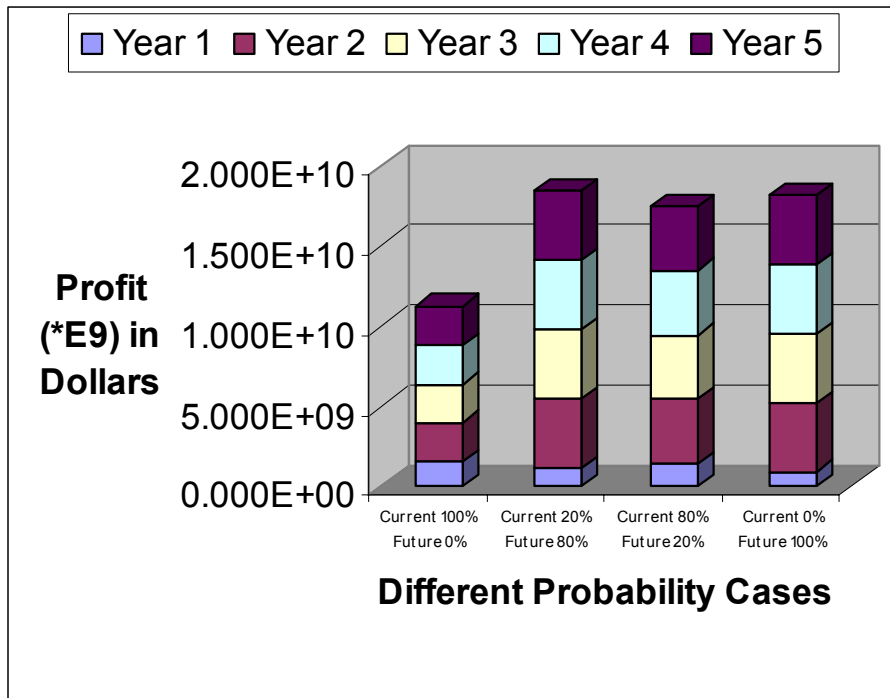
Figure 5.30: Total Profits in Different Probability Scenarios

Consider the first column of Table 5.6 where the manufacturer designs the product platform for the current market space only (probability of future extension is 0%). This is as in Scenario 1 – Product Platform Design for the Present. The profit obtained for the current market space is $1.533E+09 as obtained in Scenario 1 (LHS of Figure 5.22). Now, as discussed, the manufacturer wishes to extend his market space after the first year as it is more profitable. The profit obtained by the manufacturer for second year is the profit obtained by extending the same product platform designed for the present in Scenario 1 (RHS of Figure 5.22). The same profit continues for the next 3 years. The total profit obtained after 5 years is $1.123E+10.

Similarly, in the second column, the results in Scenario 2 are considered where the probability of the manufacturer to continue in the same market space is 20% while to extend into the adjacent space is 80%. This involves using PPCTM-RCM with the probabilities predicted by the manufacturer to be correct.

In the third column, the results in Scenario 3 are considered where the probability of the manufacturer to continue in the same market space is 80% while to extend into the adjacent space is 20%. This involves using PPCTM-RCM with the probabilities predicted by the manufacturer to be wrong. Here, the manufacturer predicted that the future market space would not be that profitable and hence he/she would continue in the same current market space for a longer time, as opposed to what actually happens.

In the fourth column, the manufacturer designs the product platform for the future market space only. This is as in Scenario 1 – Product Platform Design for the Future.

It is observed that the total profit after 5 years obtained in the second column where the probabilities predicted by the manufacturer are correct and the design is done using PPCTM-RCM is the highest profit ($1.843E+10). This helps to verify and understand the significance of the economic gain achieved by using the method.

Now, consider another manufacturer B designing a pressure vessel platform in a different situation who predicts the current extent and future extent of the market having a probability of 80% and 20% respectively. We assume that the manufacturer is correct in his prediction and the current pressure market is only profitable for 4 continuous years and he/she extends his platform in the neighboring market only in the fifth year. Hence,

he/she gains profits in the current market extent for 4 years and in the extended market space for one year.

The manufacturer has a number of options for designing the product platform in the current market space at the start of the five year period. As discussed previously in this section, he/she can use the non-augmented PPCTM and design the product platform as shown in the two cases of Scenario 1 or he/she can use the augmented PPCTM-RCM and design the product platform. All the possible cases are compared by using the results obtained in different scenarios considered till now (Table 5.7). The final profit obtained by the manufacturer after the five year period is calculated for the different scenarios.

Table 5.7: Total Profits in Different Probability Scenarios after 5 Years for Manufacturer B

	Scenario 1 - Present	Scenario 2	Scenario 3	Scenario 1 - Future
	Current 100% Future 0%	Current 20% Future 80%	Current 80% Future 20%	Current 0% Future 100%
Year 1	1.533E+09	1.123E+09	1.396E+09	8.742E+08
Year 2	1.533E+09	1.123E+09	1.396E+09	8.742E+08
Year 3	1.533E+09	1.123E+09	1.396E+09	8.742E+08
Year 4	1.533E+09	1.123E+09	1.396E+09	8.742E+08
Year 5	2.425E+09	4.328E+09	4.033E+09	4.330E+09
Total Profit After 5 Years	8.557E+09	8.820E+09	9.617E+09	7.827E+09
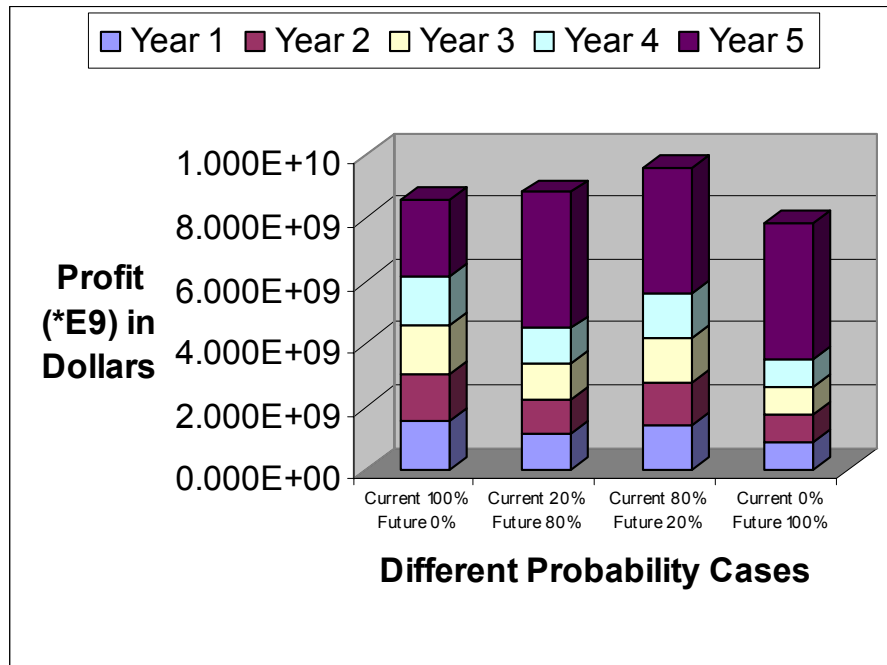
Figure 5.31: Total Profits in Different Probability Scenarios

Consider the first column of Table 5.7 where the manufacturer designs the product platform for the current market space only (probability of future extension is 0%). This is as in Scenario 1 – Product Platform Design for the Present. The profit obtained for the current market space is $1.533E+09 as obtained in Scenario 1 (LHS of Figure 5.22). Now, as discussed, the manufacturer wishes to extend his market space only after four years as it is more profitable. The profit obtained by the manufacturer for fifth year is the profit ($2.425E9) obtained by extending the same product platform designed for the present in Scenario 1 (RHS of Figure 5.22). The total profit obtained after 5 years is $8.557E+09.

Similarly, in the second column, the results in Scenario 2 are considered where the probability of the manufacturer to continue in the same market space is 20% while to extend into the adjacent space is 80%. This involves using PPCTM-RCM with the

probabilities predicted by the manufacturer to be wrong. Here, the manufacturer predicted that the future market space would be profitable and hence he/she would not continue in the same current market space for a long time.

In the third column, the results in Scenario 3 are considered where the probability of the manufacturer to continue in the same market space is 80% while to extend into the adjacent space is 20%. This involves using PPCTM-RCM with the probabilities predicted by the manufacturer to be correct.

In the fourth column, the manufacturer designs the product platform for the future market space only. This is as in Scenario 1 – Product Platform Design for the Future.

It is observed that the total profit after 5 years obtained in the third column where the probabilities predicted by the manufacturer are correct and the design is done using PPCTM-RCM is the highest profit (9.617E+09). This helps to verify and understand the significance of the economic gain achieved by using the method.

Thus from both manufacturers A and B, one observes that using PPCTM-RCM helps to design a product platform that is going to give more profit in the coming years.

5.4 VALIDITY OF THE RESULTS

The results can be validated in five ways:

1. Comparison with results of the newly designed product platform

2. Checking the constraints

3. Relation between Market Forecasts and Final Results

4. Larger marketplaces lead to larger extent of commonalization than smaller marketplaces

5. Relation of Modularity and Extent of Marketplace

1. **Comparison with Results of Product Platform with Different Probabilities-** The results is verified by considering the platform designed for the 10-40 MPa market shown in Scenario 1 - Figure 5.23. It is seen that the profit obtained by this new product platform in the after extension market is greater than the profit obtained by extension of the existing platform shown in RHS of Figure 5.22. The reason is that the new platform is designed for the bounds of this new market place, while the previous one was just an extension of the existing platform designed for a different market place. However, as discussed earlier, the problem in redesigning the product platform involves high restructuring costs.

At the same time, it is observed that the profit shown in LHS of Figure 5.22 is greater than that obtained from the results in this platform designed only for the future shown in LHS of Figure 5.23. This again helps in validating the work. The results obtained by compromising between the initial and potential market spaces will give lower profits in the latter case than when the product platform is designed especially for the potential market space.

2. **Checking the Constraints -** There are three constraints in the problem. Firstly, the range of modes at lower levels must be smaller than the range of modes at higher levels. This is seen from Table 5.2 that all the lower levels (e.g., V1) are

smaller than the higher levels (e.g., V2). Secondly, the constraint that $d_i^-, d_i^+ \geq 0$ needs to be satisfied. Again, it is observed that all the values in the Table 5.4 show that $d_i^-, d_i^+ \geq 0$. The third constraint is that $d_i^- \cdot d_i^+ = 0$. From the column of d^+, it is seen that all the values are zero. Thus, one always obtains the product of d^+ and d^- as zero.

$\Delta V3$ (m^3)	$\Delta V2$ (m^3)	$\Delta P1$ (MPa)	$\Delta V1$ (m^3)	Profit		Variance	
				d_1^-	d_1^+	d_2^-	d_2^+
20	0.5	0.5	0.5	0.106	0	0.7805	0
20	1	0.5	0.5	0.1002	0	0.7804	0
20	1	0.5	1	0.1044	0	0.7786	0
10	2	0.5	1	0.3298	0	0.6742	0
20	2.5	5	2.5	0.114	0	0.772	0
20	1	1	0.5	0.1141	0	0.774	0
10	5	1	2.5	0.368	0	0.647	0
20	2.5	2	2.5	0.1569	0	0.7498	0
20	5	2.5	0.5	0.155	0	0.75	0
20	4	10	2	0.3825	0	0.554	0

Figure 5.32: Constraint verification

3. **Relation between Market Forecasts and Final Results -** The usefulness of the results in the pressure vessel example for changing marketplaces depends on the market situation that is predicted. In Section 5.3.2, it was analyzed that in the compromise case, the profit for the current market situation ($1.123E09) is lesser than that of profit in the previous case ($1.533E09), while profit for the future market situation ($4.328E09) is higher than the profit in the previous case ($2.426E09). These results reflect the weights assigned to each of the objectives in the formulation of the compromise DSP in Section 5.2 Step 7. It was stated that the market situation is going to change quickly and the product platform needs to

be designed such that it can quickly adjust to the new market situation. So the results are obtained accordingly.

This also helps to assess the importance of proper market forecasts. Now, consider that the future potential expansion of the market is not definite and it can take more time. So by using the same results, the manufacturer is obtaining lesser profit for greater amount of time than it could have obtained by designing the platform only for the current market situation. This was observed while considering manufacturers A and B. Hence, the step of forecasting the market and assigning appropriate weights to the initial and potential marketplaces is important

4. **Larger marketplaces lead to larger extent of commonalization than smaller marketplaces -** In the pressure vessel problem of changing marketplaces, in the first case of Scenario 1, one notices that the number of products for a certain area of the market is high (subspace elements are small and hence commonalization less). In the second case of Scenario 2, the number of products for the same area of the market decreases (subspace elements are large and hence commonalization more). The difference in both the cases is that the first one is the design of a product platform for the extent of market having pressure range from 0-30MPa, while the third one is design of a product platform for the extent of market having pressure range from 0-40MPa. Hence, if the design of the product platform is for a larger extent of marketplace, the extent of commonalization of the product platform increases as compared to design of a product platform for a smaller marketplace. This trend makes sense since one cannot expect to provide large

amount of customization if the market extent to be covered by the manufacturer is large and demand does not increase in same proportion.

It is seen in Scenario 2, the number of products for the same area of the market is in between that of case 1 and 2 of Scenario 1. The reason is obvious since this scenario represents a compromise between both the other scenarios.

5. **Relation of Modularity and Extent of Marketplace** – The decision variable ΔV_3 represents the modularity of the product platform. It is observed that in Scenario 1, in the case where the platform is designed for the present market (Figure 5.20) the value of ΔV_3 is 20, while in the case where the platform is designed for the future market (Figure 5.23) the value of ΔV_3 is 10. This means that when the market space is small (Figure 5.20) modularity of pressure vessels is not desired, while when the market space is large, for volumes greater than $30m^3$ modularity is favorable. Moreover, it is also seen in Scenario 2 when a compromise is obtained between the two possible extents of market, the value of ΔV_3 is $15m^3$, which is in between the above two cases.

It makes no sense to manufacture two vessels for a small requirement of pressure and volume. However, as the requirement increases, it is much more profitable to manufacture two smaller vessels satisfying the requirement. Thus the results make sense.

5.5 A LOOK BACK AND A LOOK AHEAD

In the previous Chapter 4, the first part of the primary research question was addressed. This involved designing product platforms robust to uncertain demand and

changing design parameters. With the help of this chapter, part of the empirical and performance validity was established.

In this chapter, the last part of the primary research question is addressed.

"How should the Product Platform Constructal Theory Method be augmented so that platforms are designed strategically considering future portfolio expansions?"

The method discussed in Section 3.3 is applied to the pressure vessel example problem. It is better to use this example, since one can build on the previous chapter's problem as well as focus on the method at hand than focus on the modeling of the example. This example provides a two dimensional space of customization as a market. So it provides a good example where changes in the extent of marketplace are easily visualized. There have been many difficulties in getting the best design variables for this problem in literature. Moreover, in this case, since there are an infinite number of products to be designed, the complexity increases. Also, due to the reasons discussed in Section 4.3, one can say that this example is relevant to the problem. The empirical performance validity is demonstrated by showing that this method produces better results than a method without considering future potential expansions. The analysis and validity of the results have been discussed in Sections 5.3 and 5.4. Thus along with Chapter 4, the empirical structural and performance validity of the thesis is established.

CHAPTER VI

CLOSURE

In the thesis, the Product Platform Constructal Theory Method (PPCTM) has been augmented. The augmentations involve infusion of robustness into the method to handle uncertain demand, changing design parameters and changing extents of marketplaces. The third chapter presents the method, while the fourth and fifth chapters apply the method to examples.

In this chapter, the thesis is brought to a close. In Section 6.1, the research questions and the validity of the hypotheses are reviewed. In Section 6.2, the achievements in this thesis are discussed. The limitations of the method are noted in Section 6.3. In Section 6.4, potential avenues of future work are discussed and in Section 6.5, closing remarks are made.

	Relevance	Hypotheses
Chapter 1 Designing Product Families for Mass Customization	• Introduction, Motivation and Frame of Reference • Research Questions and Hypotheses • Organization of the thesis	Introduce
Chapter 2 The Theoretical Foundation	• Theoretical foundation of PPCTM • Detailed description of Hernandez's PPCTM • Description of constructs of potential augmentations • Theoretical structural validation	Elaborate
Chapter 3 A Method for Designing Product Platforms to Deal with Uncertain Demands, Changing Design Parameters and Changing Marketplaces	• Infuse concepts of robustness into product platform design • Use compromise DSP for strategic product platform design • Description of the method • Theoretical structural validation	Method for Implementation of Hypotheses
Chapter 4 Customized Product Platform Design in an Uncertain Environment: Pressure Vessels and Hand Exercisers	•Demonstrate that the proposed augmentation is able to deal with changes in demand and design parameters. •Empirical Structural and Performance Validity	Verify H1
Chapter 5 Customized Product Platform Design for a Changing Marketplace: Pressure Vessels	•Demonstrate that the use of compromise DSP is an effective way of strategically designing product platforms for a changing marketplace. Empirical Structural and Performance Validity	Verify H2
Chapter 6 A Look Back and A Look Ahead	•Summarize and critically review research findings •Contributions and limitations •Recommend future work •Theoretical performance validation	Summarize

Left-side section labels: Foundation / Problem Identification, Method, Hypothesis Testing, Closure

Figure 6.1: Thesis Roadmap

6.1 A LOOK BACK - WHAT HAS BEEN PRESENTED

In this section, the work presented in the thesis has been revisited through the validation square. First, the primary research questions and the hypotheses are presented. Next, the validation strategy used in the thesis to validate the hypothesis is summarized together.

6.1.1 Revisit – Research Questions and Hypotheses

Research involves posing appropriate questions, formulating answers to these questions as a set of hypotheses, and demonstrating that the proposed answers are valid. In order to build confidence in the usefulness of the proposed augmentations for meeting the challenges posed by the research questions, one must validate it. First of all the research questions and the hypothesis discussed in Chapter 1 are revisited. Then the validation strategy is reviewed.

The primary research question posed in this thesis is:

Primary Research Question

How should the Product Platform Constructal Theory Method be augmented to deal with the uncertainty in demand, changing design parameters and marketplaces?

This primary research question is addressed by investigating a set of research questions and associated hypotheses.

Q1.1: *"How should the Product Platform Constructal Theory Method be augmented in order to cope with the uncertainty and inherent changes in demand?"*

Q1.2: *"How should the Product Platform Constructal Theory Method be augmented in order to cope with the changes in design parameters?"*

Hypothesis 1: The infusion of concepts of robustness into the Product Platform Constructal Theory Method enables to design platforms that are unaffected by changes in demand and design parameters.

Q2 *"How should the Product Platform Constructal Theory Method be augmented so that platforms are designed strategically considering future portfolio expansions?"*

Hypothesis 2: Compromise DSP can be used in the Product Platform Constructal Theory Method to make tradeoffs between objectives of the initial market extent and future probable extensions.

The validation of the thesis has been done in different sections throughout the work. All the appropriate sections are brought together to review and renew the validation square in the proceeding section.

6.1.2   Validation of the Method

An overview of the validation strategy followed in this thesis is presented in Section 1.4 and observed in Figure 6.1 and 6.2. The strategy is formulated according to the validation square (Pederson et al., 2000) as shown in Figure 6.1. The theoretical structural validity, empirical structural validity, and empirical performance validity have already been discussed in sections throughout the thesis. The review of this validity and

along with discussion of the last component of theoretical performance validity is presented in this section.

Inspired by the Masters thesis of Carolyn Seepersed (Seepersed, 2002) and Christopher Williams (Williams, 2003), a similar validation format is used in the thesis.
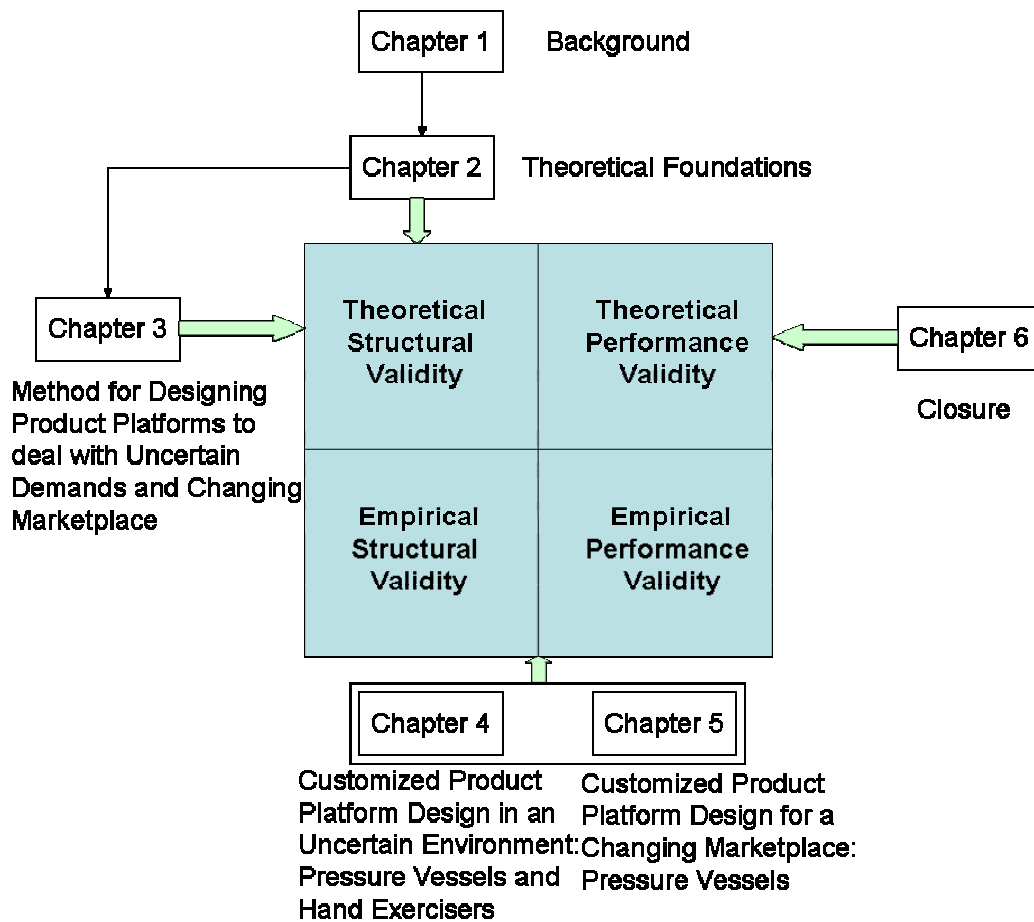
Figure 6.2: Chapters in the Validation Square

Theoretical Structural Validity	
Literature review of existing product platform design methods, detailed description of PPCTM (foundation of this work, developed by Hernandez) and the theoretical foundations of the constructs that are to be used to handle uncertain demand, changing design parameters and changing extents of marketplaces.	Chapter 2
Note down the steps needed for each of the augmentations and then provide a detailed step-by-step augmented PPCTM with implementation flowchart that alleviate all the limitations.	Chapter 3

Empirical Structural Validity	
Discuss the appropriateness of the hand exerciser problem: -Document that the abstracted ideas of the constructs presented are relevant to the application of the hand exerciser problem. -Document that the hand exerciser problem is representative of an actual problem for which the method is intended. -Document that the data associated with the hand exerciser problem can support a conclusion or conclusions for Hypothesis 1.	Chapter 4
Discuss the appropriateness of the pressure vessel problem: -Document that the abstracted ideas of the constructs presented are relevant to the application of the pressure vessel problem. -Document that the pressure vessel problem is representative of an actual problem for which the method is intended. -Document that the data associated with the pressure vessel problem can support a conclusion or conclusions for Hypothesis 1 and 2.	Chapter 4, 5

Empirical Performance Validity	
Build confidence in the usefulness of the method using the pressure vessel and hand exerciser Problems.	Chapter 4, 5

Theoretical Performance Validity	
Build confidence in the generality of the approach or in the usefulness of the approach beyond the specific example problems.	Chapter 6

Figure 6.3: Validation Strategy Related to Thesis Format

Theoretical Structural Validity

Theoretical structural validity involves the consideration of advantages, limitations, and accepted domains of application for each of the constructs constituting the theory or method as well as accepting the validity of the internal consistency of their assembly

together. This is accomplished through a literature review and a critical review of each of the building blocks considered.

There are two major decisions involved with accepting the theoretical structural validity of this work; they are explained below.

1) Are the constructs themselves valid? Have they been cited in literature before?

This question is addressed in Chapter 2 where the theoretical foundation of this work is established. Some highlights of this presentation include:

- The core focus in this work is the augmentation of the Product Platform Constructal Theory Method. In Chapter 2, firstly the hierarchic systems theory and constructal theory, building blocks of PPCTM, are discussed. Based on this discussion, Hernandez's abstraction into the realm of product family design is presented in Section 2.1.2. In Section 2.1.3, the six steps of the PPCTM are presented. The discussion of the PPCTM is ended by its critical analysis in Section 2.1.4. The limitations existing in the PPCTM such as the inability to consider uncertain demand, changing design parameters and changing extents of marketplace are presented in this section. This section also provides a series of citations documenting its previous domains of application.

- The theoretical structural validity of the constructs to augment PPCTM is discussed in Section 2.2, 2.3 and 2.4. The Robust Concept Exploration Method is presented in Section 2.2, which addresses the limitation of uncertain demand and changing design parameters. The compromise DSP, discussed in the next section also plays an important role in alleviation of the aforementioned limitations as well as the problem of changing extents of marketplaces. The idea of strategic design,

which forms the basis of considering changing extents of marketplaces while designing, is discussed in Section 2.4. Each of the constructs are discussed in detail along with its critical analysis at the end

2) Is the resulting combination of these constructs method valid? Is the combination consistent with the constructs themselves?

This question is discussed in Chapter 3 where the augmented Product Platform Constructal Theory Method is presented. Some salient points of this discussion include:

- In Section 3.1, some of the steps of RCEM are integrated into the PPCTM to present a product platform design method that is able to handle uncertain demand and changing design parameters. The RCEM has been used previously in the domain of product platform design.

- In Section 3.2, the steps to be incorporated into the PPCTM so that it is able to handle changing extents of marketplaces are discussed. Specifically, the use of compromise DSP to make a tradeoff between the current market objectives and future potential markets is presented. The application of compromise DSP in the domain of product platform design is also noted (Seepersed, 2000).

- In Section 3.3, the augmented PPCTM is presented. Each of the eight steps is explained in detail in Sections 3.4.1 through 3.4.8, wherein the new augmentations are highlighted.

- In the end in Section 3.4, the theoretical structural validity of the augmented PPCTM is provided. The validity of infusing the robust design principles and compromise DSP into the existing PPCTM is discussed.

Thus, confidence in the correctness and consistency of the constructs of the PPCTM, RCEM and the cDSP, as well as their combination to create the augmented PPCTM, is developed.

Empirical Structural Validity

Empirical structural validity involves building confidence in the appropriateness of the example problems chosen for illustrating and verifying the performance of the design method. One decision is required for the completion of this portion of the validation square: *Are the example problems appropriate?* Three steps are suggested for answering this question:

1) Are these example problems similar to the domain of which the constructs are generally accepted?

2) Do the example problems accurately represent the real world situations they attempt to portray?

3) Can the data obtained from the example problems support a conclusion with respect to the research hypotheses?

In Chapter 4, the application of the augmented PPCTM in context of uncertain demand and changing design parameter is discussed. This chapter provides empirical structural validity to the first hypotheses.

This chapter is comprised of two example problems – pressure vessel and hand exerciser. The pressure vessel problem helps to address the Q1.1, which deals with the

uncertainty and inherent changes in demand. The hand exerciser problem helps to address the Q1.2, which deals with the changes in design parameters. The chapter is divided into two parts 4.1 and 4.2. Section 4.1 deals with the pressure vessel example while the Section 4.2 deals with the hand exerciser example. Each part has five sections in it. The first section is comprised of the problem statement. Then the modeling of the example is presented. The third section helps to address the application of the augmented method to the example problem. The author in the fourth and fifth sections present the results and the analysis and validity of the results. The aforementioned questions are addressed in Sections 4.3 and 5.5 after the presentations of each of the example problems.

The pressure vessel problem helps to address the problem of uncertain demand in the design of product platforms. The pressure vessel is a classic mechanical engineering problem that has been used to validate work on product platforms.

The hand exerciser problem helps to address the problem of changing design parameters in the design of product platforms. It is a product that often requires customization. It is a challenge for manufacturers to achieve such customization economically due to the competitive nature of the product. Therefore, this design problem is representative of a product that would benefit from the application of the theory and method proposed.

Hernandez and Williams have used the same examples in their previous works including dissertation and thesis (Hernandez, 2001 and Williams, 2003). Hence, it is useful to compare the results and analysis with previous methods. If the original PPCTM is assumed to be a valid method, then the use of a similar example problem to test an augmented PPCTM is also valid.

Though the modeling of the problems looks relatively simple, the presence of a mix of continuous and discrete variables and non linear functions and models make selection of design variables a complex task. At the same time, the main focus of the thesis is to concentrate on the method than concentrating on the complex modeling of examples. Hence, the selection of the example problems is relevant.

In Chapter 5, the augmented method is applied to handle changing extents of a pressure vessel market. This involves strategically designing a line of customizable pressure vessels that will consider objectives of current as well as potential future expansions of the market. This example helps to build on the method used to design a pressure vessel platform in the previous chapter, where there was fixed market space.

The author's primary objective in this chapter is to address the second section of the primary research question and hypothesis. In Section 5.1, the problem statement is presented. In Section 5.2, some of the aspects of modeling of the pressure vessel have been discussed. Section 5.3 presents the application of the augmented method to the problem in hand. Section 5.4 is comprised of the results and its analysis. Section 5.5 helps to validate the results. In the end, a look back and ahead is made.

In Chapter 5, the application of the augmented PPCTM in context of changing marketplaces, thus providing empirical structural validity (Section 5.5) to the third hypothesis is discussed. It is better to use the pressure vessel example, since one can build on the previous chapter's problem as well as focus on the method at hand than focus on the modeling of the example. This example provides a two dimensional space of customization as a market. So it provides a good example where changes in the extent of marketplace are easily visualized. There have been many difficulties in getting the best

design variables for this problem in literature. Moreover, in this case, since there are an infinite number of products to be designed, the complexity increases.

Thus after this discussion, the augmentations proposed in Hypotheses 1, and 2, are declared empirically structurally valid.

Empirical Performance Validity

Empirical performance validity involves building confidence in the usefulness of the augmented method. It indicates the degree to which the method's purpose has been achieved. The other aim is to establish that the resulting usefulness is, in fact, a result of applying the theory/method. For example, solutions obtained with and without the construct, theory and/or method can be compared.

The empirical performance validity is established in Sections 4.3 and 5.4 for the pressure vessel and hand exerciser example problems.

- The results and their analysis of the application of the augmented PPCTM to the pressure vessel example problem are presented in Sections 4.1.4 and Section 4.1.5 respectively. The usefulness of the method is observed by comparing the results of a robust product platform and a non robust product platform. Specifically, the scenarios of product platforms without considering uncertain demand and considering uncertain demand are compared. It is seen that profit obtained is comparatively less affected despite of changing demand in the robust product platform scenario.

- The results and their analysis of the application of the augmented PPCTM to the hand exerciser example problem are presented in Sections 4.2.4 and Section 4.2.5

respectively. The usefulness of the method is observed by comparing the results of a robust product platform and a non robust product platform. Specifically, the scenarios of product platforms without considering changing design parameters and considering design parameters are compared. It is seen that cost incurred is comparatively less affected despite of changing design parameters in the robust product platform scenario.

- The results and their analysis of the application of the augmented PPCTM to the hand exerciser example problem are presented in Sections 5.3 and Section 5.4 respectively. The usefulness of the method is observed by comparing the results of the product platform considering changing extents of marketplaces with product platforms designed for only existing markets. The comparison is done with platforms designed in the initial marketplace as well as newly designed platforms for the new marketplaces. The economic gains by use of the method over a five year period are noted. The usefulness in obtaining a compromise between these two is observed in the results.

Thus, from the discussion of the usefulness of the augmentations proposed in Hypotheses 1, 2, and 3, it is concluded that the augmented PPCTM is valid in terms of empirical performance, for the chosen example problems.

Theoretical Performance Validity

Theoretical performance validity involves building confidence in the generality of the augmented method and accepting it as useful beyond the example problems. This is

done by showing that the example problems are representative of a general class of problems and that the augmented method is useful for these problems.

The general class of problems for which the constructs of this augmented PPCTM are valid are defined by the following characteristics:

For product platform development:

- Design problems having uncertain demand

- Design problems considering changing design parameters

- Product platform development problems involving changing extents of marketplaces

- Design problems involving multiple conflicting objectives

- Design problems involving a market of non-uniform demand

- Product platform development problems that integrate systematically and coherently the utilization of multiple approaches for managing product variety, such as modular design, dimensional customization, parametric scaling, etc.

- Product platform development in which the products are functionally similar, but with different performance requirements

- Product platform development problems involving a distinct tradeoff for commonalizing a design parameter for the sake of performance

- It can be applied to any number of customizable specifications.

As part of the requirements of establishing empirical structural validity, it has been shown that the example problems are representative of a class of general problems with the preceding characteristics. In order to establish empirical performance validity, it

has been shown that the methods are useful for the example problems. By induction, it is argued here that the methods are useful for a general class of problems identified by the preceding characteristics.

Thus in this section, each portion of the validation square has been reviewed. It is therefore asserted that the primary research question has been answered. *The Product Platform Constructal Theory Method is augmented in order to deal with the uncertainty in demand, changing design parameters and marketplaces. This has been achieved by the infusion of concepts from Robust Concept Exploration Method and the use of compromise Decision Support Problem.*

6.2   LIMITATIONS

While the goals established at the beginning of this thesis are close to being accomplished, there are a number of limitations of this work that must first be addressed. A large portion of this critical analysis is presented throughout this thesis, specifically in the application of the augmented PPCTM to the two example problems in Chapters 4 and 5. These criticisms are summarized again here. It should be noted that many of the limitations of this research will be transformed into avenues for future work in the following section.

Several limitations of this research are derived from the series of assumptions listed at the beginning of the description of each of the example problems:

- In the application of the augmented PPCTM to the development of product platforms, only scalable product components are considered. In other words,

variations of product functionality are not considered. One possibility that can be investigated is applying the proposed theory and method independently to functional modules. A way to represent this would be that each functional module would have its own market space for which an individual module platform is designed and the overall market space for the complete product would be the union of all the market spaces of the modules. With respect to the examples in the thesis, only a few modes of managing variety are modeled. Specifically, dimensional customization, parametric scaling, and modular combination are the only modes considered. Functional based, conventional modularity, etc. are not considered in the examples. Also, it is important to note that since in the method a tradeoff is obtained between commonality and performance, every mode needs to consider both these aspects or else that mode is ignored in the method.

- The limitation of the work is that if the number of future probable extensions is more, the method becomes computationally extensive. It is assumed that the probable future extensions are limited to a small number. So if it is not possible to narrow down the probable extensions to a small number, the computational analysis increases since now one has to consider a couple of more objectives simultaneously.

- A limitation of the work is in the approximations of the examples to adapt to the research. The example problems are modeled as effectively as possible while keeping the size of the resultant approximation in check.

It is the author's opinion that these limitations provide an effective way to explore the three principle tasks in the completion of this research and do not reduce the overall

utility and validity of the augmented PPCTM.   In addition to these limitations, there are

some areas that the author considers potential future work, which is discussed in the next

section.

6.3   ACHIEVEMENTS AND CONTRIBUTIONS

The primary research question stated in Chapter 1 is:

> *How should the Product Platform Constructal Theory Method be augmented to*
> *deal with the uncertainty in demand, changing design parameters and marketplaces?*

From the primary research question as well as the corresponding sub-questions asked, it

is clear that there are three core objectives of this research.

1.   Deal with uncertain demand

2.   Cope with changing design parameters

3.   Handle changing extents of marketplaces

Based on addressing these objectives, there are a number of contributions made in the

thesis:

1.   *Deal with uncertain demand and changing design parameters:* The method helps

to enable the designer to deal with uncertain demand and changing design

parameters. Robust design principles have been infused into the Product Platform

Constructal Theory Method so that the resulting product platforms have minimum

variance with fluctuations in demand and design parameters over time.   The

proposed method is able to handle the inherent uncertainty present in markets of

customizable products. This method is used to design product families that

improve the performance objectives and minimize their variation to market and

design parameter uncertainties. The integration of RCEM enables addressing individually the issues of bringing the mean on target and reducing variation. This provides designers more flexibility to make decisions based on different robust design criteria and help them focus on major hurdles and make improvements. The authors have also explored the sensitivity of the configuration of the proposed platform design to different weightings of the objectives. It has been shown that the use of the PPCTM gives appropriate results according to the manufacturer's preference for "moving the mean to target" and "minimizing variation."

2. *Handle changing extents of marketplaces:* The authors propose a method to strategically design product platforms that not only consider the objectives of the initial market space but also those of the future potential market expansions. This helps to develop a product family life cycle instead of getting restricted to a fixed extent of marketplace. This is especially useful in today's world where changes in the market are very rapid. This method makes it possible to obtain the best possible performance values of the product platforms in the initial as well as the future expansions. The use of the compromise DSP helps in making tradeoffs between the conflicting objectives of the initial product portfolio and the future potential product portfolios. Hence one is able to obtain specifications of the initial product platform that will also give favorable results for future expansions. The need of restructuring the initial product platform or redesigning the entire product platform again is avoided.

3. *Product Platform Design Method:* As discussed in Section 1.2.2, although there are a number of methods to design product platforms, all but a few suffer from a

number of limitations. After the work in this thesis, it is the author's opinion that PPCTM is able to overcome all of the limitations and is able to design product platforms with almost all of the needed characteristics discussed in Section 1.2.2. Moreover, from the augmentations suggested, the author has also noted down the key lessons learned in the domain of product platforms.

6.4   FUTURE WORK

- **Functional module capability –** As discussed in the limitations, using PPCTM the designer does not consider variety with respect to functional module capabilities. In the case of modules, since a module in one product is a module in another product, commonalization means that the module slot must be compatible with each other. Functional compatibility means the same interface of the module against the other part of a product, technologically and geometrically. There are two important questions that need to be addressed.

    1. Right now, PPCTM considers the commonalization of design parameters that help provide variety. In the case of modular architecture, one needs to consider commonalization of modules in different products. Here, one considered different modes of managing variety in a hierarchical manner, in the case of modular architecture, there would be a functional hierarchy (main functions, sub functions, primitive functions) and a corresponding module hierarchy (main system, sub systems). How could one use constructal theory concepts to define and commonalize the modules at each level of the hierarchy?

2. Each functional module would have its own market space. The overall market space for the complete product platform would be the union of all the market spaces of individual modules. So an important factor for consideration while deciding the modules of a platform would be maximization of this union of market spaces.

- **Functional Domain to Physical Domain** – In Section 4.2.5, the limitation of PPCTM of not having a well defined procedure for allocation of modes of managing variety has been discussed. Moreover, it does not have a defined procedure for getting these modes. A possibility of future work is in extending PPCTM to explore the system architecture level of design. Apart form the scaling based approach in developing platforms; it is important how product architecture is built based on change from functional domain to physical domain. Every product concept that is generated at the conceptual level needs to be evaluated based on its feasibility on a platform level. There is no point in developing the optimal concept for a single product when a product platform needs to be developed. The modes of managing variety will vary based on the concept selected. So instead of defining the concept and working on a limited number of modes, one needs to develop a method that evaluates different concepts based on different hierarchies of modes possible for every concept and determine the extent of application of modes for the hierarchy selected.

- **Qualitative to Quantitative Analysis -** There is a need to convert the subjective and qualitative data of the customers into quantitative and useful data. This will help to define the market space more accurately and thus the design of product

platform would be according to needs. This in turn would help to predict the potential expansions of market spaces more accurately. Approaches like the conjoint analysis can be explored in this area. Right now, it is assumed that the market research has already been done and is not a part of the development of the method. Integrating such analysis into the method could help in designing platforms closer to user needs.

6.5   CLOSING REMARKS

In this section, I will like to think loud – about the method discussed, its future and some other parting ideas.

The first question that arises in the reader's mind is that "Why should I use this method for product platform design?" It is necessary to note some features of the PPCTM that makes it really different from the existing platform design methods. First and foremost, it is the only method that enables the designer to synthesize multiple modes of managing product customization. Normally, other methods only have the platform designed on one or two modes of managing variety. Secondly, this method makes it possible to have a tradeoff between commonality and performance. Here, at the beginning stage, one does not know how many products need to be provided, which is an implicit assumption in some other methods. The abstraction of the concepts of constructal theory in this field of product platform is a novel and effective way to tackle the problems. The augmentations of the PPCTM have made it possible to consider non-uniform and uncertain demand, changing design parameters and extents of marketplaces. Through PPCTM, the novel idea of process platforms has also been introduced. No other

method is able to consider all these different aspects of product platform design. No other method provides such a systematic and methodical procedure for product platform design.

The future of PPCTM is bright. The above-mentioned advantages really provide a vector of differentiation to this method. I would be excited to see this method being implemented in the actual design of a product. Only then will the exact problems that the designer faces would come forward and one could improve further. I find this method not only to give great results in the product platform field but with some modifications I foresee that this method can be really useful in other fields also. Wherever there is a need to commonalize things, this constructal theory based approach can be really useful. For example, in project management, one could very well commonalize projects so that allocation of resources can be much simpler. Instead of allocating resources to individual projects, it will always be better to commonalize projects and allocate resources to the entire group. I am excited about the future of the method in such diverse applications.

APPENDIX A

JAVA CODE FOR THE PRESSURE VESSEL EXAMPLE PROBLEM

In this appendix, the computer program for the pressure vessel example is presented. The program is written in Java. A brief description of the classes and functions written is followed by the actual code.

The solution algorithm used for this example is as explained in Section 4.1.3, which is modified from Williams, 2003. A brief description of the classes and functions are as follows:

Classes

cudPV   This is the main class of the program for uncertain demand. From this class different functions are called. The solution algorithm is implemented in this class.

cMS   This class deals with all the details of developing a market space. This involves discretization of the market space.

cDetailsOP   This class helps to write the output. i.e., detailed specifications of the product families into a file.

cResultsOP   This class is used to display the results of the product platform having lowest deviation.

ccmPV	This is the main class for the changing extent of market program. From this class different functions are called. The solution algorithm is implemented in this class.
Functions	
Cost	To calculate the cost of a pressure vessel
normProf	To calculate the normalized profit for the entire market space
normVar	To calculate the normalized variance for the entire market space
Rmin	To find out the minimum radius given a volume using the bisection method root-finding technique
fCheck	To check if one number is a multiple of other
z	To calculate the deviation function helping to make compromise
updateR	To print $\Delta V3$, $\Delta V2$, $\Delta V1$ and $\Delta P1$ of platforms in the file
updateR2	To print profits, variances and deviations of platforms in the file
updateD	To print details of each node of the product platforms in the file
updateDp	To print the final results in the file
close	To close the file
estNodes	To establish nodes in the market space
DemandDist	To define the demand distribution of the market
calcPrice	To calculate the selling price of the vessel
getTDemand	To find the total demand in the market
getnarray	To return the array containing nodes
XDim	To return the volume dimension of the market space
YDim	To return the pressure dimension of the market space

NodeDim	To return the number of nodes in the market space
initCost	To initialize the cost of each node
initVmax	To initialize the maximum volume of each node for first stage
initVmax1	To initialize the maximum volume of each node for second stage
initPmax	To initialize the maximum pressure of each node
initRmin	To initialize the radius of each node

cudPV

```
/////////////// cudPV (Class Uncertain Demand Pressure
Vessel)///////////////////
import java.lang.*;
public class cudPV {

     ////////Initialization/////////
     double dP1compr = 0.0;
     double dV2compr = 0.0;
     double dV1compr = 0.0;
     double dV3compr = 0.0;
     double Profitcompr = 0.0;
     double Varcompr = 0.0;
     double Dev= 0.0;

     /////////Bounds////////////
     double Tl = 6.35; //max possible thickness
     double Tu = 76.2; //min possible thickness (multiples of 6.35)
     double Lu = 7.0; //max possible length
     double Ru = 4.0; //min possible radius
     double pi = java.lang.Math.PI;

     public static void main (String[] args)
     {
          new cudPV();
     }

     ///////cudPV constructor//////
     public cudPV()
     {
          double averageprofit = 0.0; //temperory variable to find
best platform
               //double averagev=1E30;
          double averageu=100000.0; //temperory variable to find best
platform

          ////Market space bounds and inner resolution
          double res = 0.5; //resolution of discretization
          double vor = 10.0; //origin of v-axis - x dimension
```

```
            double vlen = 20; //v-length of market
            double por = 10.0; //origin of v-axis - x dimension
            double plen = 20; //p-length of market

            //call MS class constructor to create node array
            cMS MS = new cMS(res, vor, vlen, por, plen);

            //call cDetailsOP and cResultsOP class constructor for
output
            cDetailsOP DetailsOP = new cDetailsOP(res);
            cResultsOP ResultsOP = new cResultsOP(res);

            /*Start with dP1, go through different values of dP1,
            each node is assigned the max pressure according to the
            discretization of P-axis. From value of max pressure, find
            minimum thicknesses for each node. Then start iterating
            different values of dV2 which is for finding out the
            radius. Then by iterating values of dV1 the length is
            determined. In the end, dV3 is also iterated to find out
            the cutoff point for combination. From the different values
            obtained, find the profit of each node. Go on storing
            profits larger than previous iteration and print and send
            to output file.
            */

            double dP1=0; //size of first-stage pressure element
            double P2=0;      //largest pressure of the considered
space
            double narray[][] = MS.getnarray();
            int Y = 0;
            int X = 0;
            int T = 0;
            Y = MS.YDim();
            X = MS.XDim();
            T = MS.NodeDim();

            //start going through dP1
            for (int i=1; i<((Y/2)+1) ; i++)
            {
                P2 = narray[i*X][2];
                dP1 = P2 - narray[0][2];

                //check to see if dP1 is a multiple of dP2
                int dP1factor = fCheck(plen, dP1);
                if (dP1factor != 1)
                {
                int dP1node = i; //number of nodes in dP1
                //number of elements in P direction
                int P1element = (Y-1)/dP1node;
                double Pmax=0;
                //Assign each node its max pressure
                narray[0][6] = P2;
                narray[0][10] = 1.0;
                for (int a=0; a<P1element; a++)
                {
                int PmaxNode = X*dP1node + a*dP1node*X;
                Pmax = narray[PmaxNode][2];
```

310

```
narray[PmaxNode][6] = narray[PmaxNode][2];
narray[PmaxNode][10] = 1.0;
//assign nodes the pressure below Pmax line
int d = PmaxNode;
while (d > 0)
{
d = d - X;
if (narray[d][10] != 1.0)
{
narray[d][6] = Pmax;
narray[d][10] = 1.0;
}
else {;}
}
}//end of Pmax loop
//assign nodes on horizontal line with Pmax
int b = 0;
while (b < (T - X)+1)
{
for(int e=1; e<X; e++)
{
narray[b+e][6] = narray[b][6];
narray[b+e][10] = 1.0;
}
b = b + X;
}

double dV2=0;
double V2h=0;

for (int h=1; h<((X/2)+1); h++)
{
      V2h = narray[h][1];
      dV2 = V2h - narray[0][1];
      int dVnodeh = h; //number of nodes in dV1

      //check to see if dV is a multiple of x dimension
      int dVhfactor = fCheck(vlen, dV2);
      if (dVhfactor != 1)
      {
      //number of elements in V direction
      int Velementh = (X-1)/dVnodeh;

      //assign each node its maximum Volume
      double Vmaxh=0;
      double Rminh=0;
      for (int m=1; m<Velementh+1; m++)
      {
      Vmaxh = narray[m*dVnodeh][1];
      Rminh = Rmin(Vmaxh);
      //assign nodes with max volume
      for (int n=0; n<((m*dVnodeh)+1); n++)
      {
      if(narray[n][9]!=1.0)
      {
      narray[n][7] = Rminh;
      narray[n][10] = 1.0;
```

311

```
                }
                else {;}
                //assign nodes above 1st line
                for(int p=1; p<Y; p++)
                {
                if(narray[n+p*X][9]<1.0)
                {
                narray[n+p*X][7] = Rminh;
                narray[n+p*X][9] = 1.0;
                }
                else{;}
                }
                }
                }//end of Vmax loop

        double dV1; //size of first-stage volume element
        double V2; //largest Volume of the considered space

        MS.initVflag();

        //go through dV1s
        for (int j=1; j<((X/2)+1); j++)
        {
                V2 = narray[j][1];
                dV1 = V2 - narray[0][1];
                int dVnode = j; //number of nodes in dV1

                //check to see if dV is a multiple of the v-dimension
                int dVfactor = fCheck(dV2, dV1);
                if (dVfactor != 1)
                {
                //number of elements in V direction
                int Velement = (X-1)/dVnode;

                //assign each node its max Volume
                double Vmax=0;
                double Rmin=0;
                for (int z=1; z<Velement+1; z++)
                {
                Vmax = narray[z*dVnode][1];
                Rmin = Rmin(Vmax);
                //assign nodes with Vmax
                for (int y=0; y<((z*dVnode)+1); y++)
                {
                if(narray[y][9]!=1.0)
                {
                narray[y][5] = Vmax;
                narray[y][9] = 1.0;
                }
                else {;}
                //assign nodes above v axis line
                for(int x=1; x<Y; x++)
                {
                if(narray[y+x*X][9]<1.0)
                {
                narray[y+x*X][5] = Vmax;
                narray[y+x*X][9] = 1.0;
```

312

```
                    }
                    else{;}
                    }
                    }
                    }//end of Vmax loop

                    /*Start iterating through dV3. Assume cutoff point
and recalculate
                    profit considering that many vessels. Find the best
combination of
                    different parameters
                    */
                    double dV3; //size of third space element
                    double V3; //cutoff point
                    //first try when there is no dV3 - no cutoff
                    dV3 = vlen;
                    //calculate cost for each node
                    for (int n=0; n<T; n++)
                    {
                    double maxV = narray[n][5];
                    double maxP = narray[n][6];
                    double V = narray[n][1];
                    double R = narray[n][7];
                    double Lo = (maxV / (pi*R*R)) - ((4.0/3.0)*R);
                    double L = (V / (pi*R*R)) - ((4.0/3.0)*R);
                    double cost = Cost(maxP, V, R, L, Lo);
                    narray[n][4] = cost;
                    }

                    //find avg profit of entire space
                    double newprofit = 0;
                    double totalDemand = 0;

                    double de[];
                    double pr[];
                    double newv=0.0;
                    de=new double[5];
                    pr=new double[5];
                    double v=0.0;
                    double unewv=0.0;
                    double unewprofit1=0.0;
                    double combou=0.0;
                    double dunewv=0.0;
                    double dunewprofit1=0.0;

                    for (int q = 0; q<T; q++)
                    {
                    //demand * (selling price - cost)
                    //provide noise
                    de[0]=narray[q][3];
                    de[1]=0.8*narray[q][3];
                    de[2]=0.9*narray[q][3];
                    de[3]=1.1*narray[q][3];
                    de[4]=1.2*narray[q][3];

                    pr[0]=de[0]*(narray[q][8] - narray[q][4]);
```

```
for(int g=1; g<5; g++)
{
        pr[g]=de[g]*(narray[q][8] - narray[q][4]);
        v=v+((pr[g]-pr[0])*(pr[g]-pr[0]))/3;

}

newv=newv+v;


newprofit = newprofit + (narray[q][3]*(narray[q][8] -
narray[q][4]));
totalDemand = narray[q][3] + totalDemand;
}
// CEquip for each dV1
double CEquip = 0;
for (int p=1; p<(Velementh+1); p++)
{
double rad = narray[p*dVnodeh][7];
CEquip = CEquip + (500000 + (50000*rad));
}

//Cost of each Lo and each dP1
double Locost = 250*Velement+100*P1element;
double newprofit1 = newprofit - CEquip - Locost;

//divide by total demand of space to take average
newprofit1 = newprofit1 / totalDemand;

//unewprofit1 = normProf(newprofit1, unewprofit1);
//unewv = normVar(newv, unewv);

unewprofit1=newprofit1/4700;

unewv=1E16/newv;

if((unewprofit1<0)||(unewprofit1>1))
        unewprofit1=0;

if((unewv<0)||(unewv>1))
        unewv=0;

dunewprofit1=1-unewprofit1;
dunewv=1-unewv;

//combou = z(0.5, 0.5, dunewprofit1, dunewv)
combou=0.5*dunewv+0.5*dunewprofit1;


if (combou < averageu)
{
averageu = combou;
dP1compr = dP1;
dV2compr = dV2;
dV1compr = dV1;
dV3compr = vlen;
```

314

```
                    Profitcompr = newprofit1;
                    Varcompr = newv;
                    Dev= averageu;

                    //output to file
                    DetailsOP.updateD(dV3, dV2, dV1, dP1, narray,
MS.NodeDim());

                    DetailsOP.updateDp(newprofit1, newv, averageu);
                    }


                    ResultsOP.updateD(dV3, dV2, dV1, dP1, newprofit,
CEquip, totalDemand, newprofit1, newv, unewprofit1, unewv, combou);


                    //iterate through dV3s
                    for (int c=((X/2)+1); c<X; c++)
                    {
                    int dV3node = c;
                    V3 = narray[dV3node-1][1];
                    dV3 = V3 - narray[0][1];

                    //check to see if dV3 is a multiple of dV1
                    int dV3flag = fCheck(dV3, dV2);
                    if (dV3flag > 0)
                    {;}
                    else //if it is a multiple iterate
                    {

                    //calculate cost for each node
                    for (int n=0; n<MS.NodeDim(); n++)
                    {
                    double V = narray[n][1];
                    double R = narray[n][7];
                    double maxV = narray[n][5];
                    double maxP = narray[n][6];
                    double Lo = (maxV / (pi*R*R)) - ((4.0/3.0)*R);
                    double L = (V / (pi*R*R)) - ((4.0/3.0)*R);
                    double cost = Cost(maxP, V, R, L, Lo);
                    narray[n][4] = cost;


                    }

                    for (int d = dV3node; d<X; d++)
                    {
                    for (int e=0; e<Y; e++)
                    {
                    int node = d/2 + (e*X);
                    //location of half of current node
                    narray[(d+(e*X))][4] = 2 * narray[node][4];
                    }
                    }

                    //find average profit

                    newprofit = 0;
```

```
                        totalDemand = 0;
                        v=0;
                        newv=0;
                        for (int q = 0; q<T; q++)
                        {

                        de[0]=narray[q][3];
                        de[1]=0.8*narray[q][3];
                        de[2]=0.9*narray[q][3];
                        de[3]=1.1*narray[q][3];
                        de[4]=1.2*narray[q][3];

                        pr[0]=de[0]*(narray[q][8] - narray[q][4]);

                        for(int g=1; g<5; g++)
                        {
                                pr[g]=de[g]*(narray[q][8] - narray[q][4]);
                                v=v+((pr[g]-pr[0])*(pr[g]-pr[0]))/3;
                        }

                        newv=newv+v;

                        //demand * (selling price - cost)
                        newprofit = newprofit + (narray[q][3]*(narray[q][8] -
narray[q][4]));

                        totalDemand = narray[q][3] + totalDemand;
                        }


                        //CEquip for each dV3
                        CEquip = 0;
                        int o = 1;
                        int onode = dVnode;
                        while (onode < dV3node)
                        {
                        double rad = narray[onode][6];
                        CEquip = CEquip + (500000 + (50000*rad));
                        o++;
                        onode = o*dVnode;
                        }

                        newprofit1 = newprofit - CEquip - Locost;

                        //divide by total demand of space to take average
                        newprofit1 = newprofit1 / totalDemand;

                        //unewprofit1 = normProf(newprofit1, unewprofit1);
                        unewprofit1=newprofit1/4700;
                        //unewv = normVar(newv, unewv);
                        unewv=1E16/newv;


                        dunewprofit1=1-unewprofit1;
                        dunewv=1-unewv;

                        //combou = z(0.5, 0.5, dunewprofit1, dunewv)
                        combou=0.5*dunewv+0.5*dunewprofit1;
```

316

```
                    if (combou < averageu)
                    {
                    averageu = combou;
                    dP1compr = dP1;
                    dV2compr = dV2;
                    dV1compr = dV1;
                    dV3compr = vlen;
                    Profitcompr = newprofit1;
                    Varcompr = newv;
                    Dev= averageu;
                    //output file
                    DetailsOP.updateD(dV3, dV2, dV1, dP1, narray, T);
                    DetailsOP.updateDp(newprofit1, newv, combou);
                    }

                    ResultsOP.updateD(dV3, dV2, dV1, dP1, newprofit,
CEquip, totalDemand, newprofit1, newv, unewprofit1, unewv, combou);


                    MS.initCost();
                    }//end of else in dV3 iteration

                    }//end of dV3 iteration

                    //initialize flags for next iteration

                    MS.initVmax();
                    MS.initVflag();

             } //end of else of dV1
             }//end of dV1


             MS.initVflag();

             }//end of else of dV2
             } //end of dV2
             MS.initPmax();
             MS.initPflag();

             }//end of else of dP1
             }//end of dP1

             //final results
             DetailsOP.updateR(dV3compr, dV2compr, dV1compr, dP1compr);
             DetailsOP.updateR2(Profitcompr, Varcompr, Dev);

             //close output files
             DetailsOP.close();
             ResultsOP.close();

             } //end of cudPV() constructor


        public int fCheck(double n, double m)
```

```java
        {
                Double x = new Double(n/m);
                int y = x.intValue();
                double z = y+0.0;
                double fractioncheck = (n/m) - z;
                int flag = 0;
                if (fractioncheck < 0 || fractioncheck > 0)
                {
                        flag = 1;
                }
                return flag;
        }//end of fCheck()


        public double Cost(double xP, double xV, double xR, double xL,
double xLo)
        {
                double P = xP; //Pressure, MPa
                double V = xV; //Volume, m^3
                double R = xR; //Radius, m
                double L = xL; //Length, m
                double Lo = xLo; //Raw Length, m

                //---- Constants ----//
                double sigy = 1077; //Yield strength, MPa
                double dens = 7800; //density, kg/m^3
                double Cs = 0.8; //processed shell stell cost, $/kg
                double Ch = 2.0; //cost of forged steel for head, $/kg
                double Cp = 0.3; //cost of raw steel plate, $/kg
                double Cw = 15; //cost of welding material, $/kg

                double Ts = (R*1000)* (P / (sigy - (0.6*P))); //shell
thickness, mm
                double Th = (R*1000)* (P / ( (2*sigy) - (0.2*P) )); //head
thickness, mm

                double CWaste = 2 * pi * dens * Cp * (Ts/1000) * R * (Lo -
L); //cost of wasted material, $
                double CMat = 2 * pi * dens * ( (Cs*R*(Ts/1000)*L) +
(Ch*R*R*(Th/1000)) ) + CWaste; //material cost, $
                double VLWeld = (4.0/9.0) * pi * (Ts/1000) * (Ts/1000) * L;
//volume of long weld, m^3
                double CLongWeld = VLWeld * dens * Cw; //cost of long weld
                double VCWeld = (8.0/9.0) * pi * pi * (Ts/1000) * (Ts/1000)
* R; //volume of circ. weld, m^3
                double CCircWeld = VCWeld * dens * Cw; //cost of circular
weld, $
                double CWeld = CLongWeld + CCircWeld; //total cost of weld

                double CTotal = CMat + CWeld; //total cost excluding
equipment cost, $
                return CTotal;
        } //end of Cost


        public double Rmin(double xV)
        {
```

```java
            //find the minimum feasible radius for a vessel
            //with given Volume
            double V = xV;
            double Verr = 0.0; //used to compare with given V
            double R1 = 0.0;
            double R2 = Ru;
            double Rmin = 0.0; //minimum radius
            double error = 1.0; //error used for testing convergence
            //iterate through volume quation with bisection method
            for(int i=0; i<30; i++)
            {
                    Rmin = (R1+R2)/2;
                    Verr = (4/3)*pi*(Rmin*Rmin*Rmin) + pi*Rmin*Rmin*Lu;
                    error = java.lang.Math.abs(Verr - V);
                    if (error < 0.01) { break; }
                    if (Verr < V)
                    {
                            R1 = Rmin;
                    }
                    else
                    {
                            R2 = Rmin;
                    }
            }
            return Rmin;
    }//end of Rmin

    public double normProf(double newprofit1, double unewprofit1)
    {
            unewprofit1=newprofit1/4700;
            return unewprofit1;
    }

    public double normVar(double newv, double unewv)
    {
            unewv=1E16/newv;
            return unewv;
    }

    public double z(double w1, double w2, double dunewprofit1, double
dunewv)
    {
            double zcombou=0.0;
            zcombou = w1*dunewprofit1 + w2*dunewv;
            return zcombou;
    }


} //end of cudPV
```

cMS

```
//////////cMS///////Class Market Space////////


 import java.lang.Number.*;
```

```
public class cMS
{

    ////////Definition///////
    double xor; //x origin
    double yor; //y origin
    double lenx; //length of x axis
    double leny; //length of y axis
    double res; //resolution variable

    //Initialization
    int xint = 0; //dimension of x-axis array
    int yint = 0; //dimension of y-axis array
    double narray[][]; //array to store data of each node

    //constructor cMS
    public cMS(double dres, double dxor, double dlenx, double dyor,
double dleny)
    {
        res = dres;
        xor = dxor;
        lenx = dlenx;
        yor = dyor;
        leny = dleny;

        estNodes();  //establishes number of nodes
        //rows = nodes
        //columns = xloc, yloc, demand, cost, maxV, maxP, Rmin,
Price, Vflag, Pflag, Rflag

        narray = new double[xint*yint][12];

        narray[0][1] = xor;
        narray[0][2] = yor;

        //assign narray with x and y locations
        for (int k=1; k<(xint*yint); k++)
        {
            for (int i=1; i<xint; i++)
            {
                narray[k][1] = narray[k-1][1] + res;
                narray[k][2] = narray[k-1][2];
                k++;
            }
            if (k<(xint*yint))
            {
                narray[k][1] = xor;
                narray[k][2] = narray[k-1][2] + res;
            }
        }

        DemandDist(); //Concentric square

        calcPrice(); //calculate selling price

    } //end constructor cMS()
```

```java
public void estNodes()
{
        //find number of nodes
        Double x = new Double (lenx / res);
        Double y = new Double (leny / res);
        xint = x.intValue() + 1;
        yint = y.intValue() + 1;

        double xd = xint+0.0;
        double yd = yint+0.0;
        double decimaltestx = (lenx/res) - xd;
        double decimaltesty = (leny/res) - yd;

}//end


public void DemandDist()
{
        //assign appropriate demand scenario by putting in narray
array

        //Demand Scenario
        double D1 = 150.0; //demand level 1 of 3
        double D2 = 250.0; //demand level 2 of 3
        double D3 = 400.0; //demand level 3 of 3

        double x1 = 14; //1st cut-off for D1 on x-axis
        double x2 = 18;
        double x3 = 22;
        double x4 = 26;
        double y1 = 14; //1st cut-off for D2 on y-axis
        double y2 = 18;
        double y3 = 22;
        double y4 = 26;

        for (int k=0; k<(xint*yint); k++)
        {
        if (narray[k][1] < (xor + lenx + 1.0)  && narray[k][2] <
(yor + leny + 1.0))
                {
                        narray[k][3] = D1;
                }
        if (narray[k][1] < x4 && narray[k][2] < y4 && narray[k][1]
> x1 && narray[k][2] > y1)
                {
                        narray[k][3] = D2;
                }
        if (narray[k][1] < x3 && narray[k][2] < y3 && narray[k][1]
> x2 && narray[k][2] > y2)
                {
                        narray[k][3] = D3;
                }
        }
} //end of DemandDist
```

```
////calcPrice()////
public void calcPrice()
{
      //Price = $x*V + $x*P
      for (int i=0; i<(xint*yint); i++)
      {
            narray[i][8] = 255*narray[i][1] + 255*narray[i][2];
      }
} //end of calcPrice()



public double getTDemand()
{
      double tDemand = 0.0;
      for (int i = 0; i<(xint*yint); i++)
      {
            tDemand = tDemand + narray[i][3];
      }
      return tDemand;
} //end of getTDemand()


public double[][] getnarray()
{
      return narray;
} //end of getnarray



public int XDim()
{
      return xint;
}//end XDim


public int YDim()
{
      return yint;
} //end of YDim



public int NodeDim()
{
      return xint*yint;
} //end of NodeDim()


public void initCost()
{
      //initializes the cost column in the array
      for (int i=0; i<(xint*yint); i++)
      {
            narray[i][4] = 0.0;
      }
}//end of initCost()
```

```java
public void initVmax()
{
   //initializes the maxV column in the array
      for (int i=0; i<(xint*yint); i++)
      {
            narray[i][5] = 0.0;
      }
}//end of initVmax()




public void initVmax1()
{
   //initializes the maxV column in the array
      for (int i=0; i<(xint*yint); i++)
      {
            narray[i][12] = 0.0;
      }
}//end of initVmax1()




public void initPmax()
{
   //initializes the maxP column in the array
      for (int i=0; i<(xint*yint); i++)
      {
            narray[i][6] = 0.0;
      }
}//end of initPmax()


public void initRmin()
{
   //initializes the Rmin column in the array
      for (int i=0; i<(xint*yint); i++)
      {
            narray[i][7] = 0.0;
      }
}//end of initRmin()


public void initR()
{
   //initializes the R column in the array
      for (int i=0; i<(xint*yint); i++)
      {
            narray[i][8] = 0.0;
      }
}//end of initR()


public void initVflag()
{
```

```
        //initializes the Vflag column in the array
            for (int i=0; i<(xint*yint); i++)
            {
                    narray[i][9] = 0.0;
            }
      }//end of initVflag()



      public void initPflag()
      {
         //initializes the Pflag column in the array
            for (int i=0; i<(xint*yint); i++)
            {
                    narray[i][10] = 0.0;
            }
      }//end of initPflag()



      public void initRflag()
      {
         //initializes the Rflag column in the array
            for (int i=0; i<(xint*yint); i++)
            {
                    narray[i][11] = 0.0;
            }
      }//end of initRflag()

 }//end cMS


cDetailsOP


 import java.io.*;

 public class cDetailsOP
 {
      BufferedWriter out;

      public cDetailsOP(double res)
      {
            try
            {
                    out = new BufferedWriter(new
FileWriter("NodeDetails.txt"));
                    out.write("\n");
            }
            catch (IOException e)
            {
                    System.err.println("FileStream error " + e);
            }
      }//end of cDetailsOP constructor

      public void updateR(double dV3, double dV2, double dV1, double
dP1)
      {
```

324

```java
        try
        {
                out.write("dV3 = " + dV3 + "\n");
                out.write("dV2 = " +    dV2 + "\n");
                out.write("dV1 = " + dV1 + "\n");
                out.write("dP1 = " + dP1 + "\n");
        }
        catch (IOException e)
        {
                System.err.println("FileStream error " + e);
        }
    }

    public void updateR2(double Profit, double Variance, double Dev)
    {
            try
            {
                    out.write("Profit = " + Profit + "\n");
                    out.write("Variance = " + Variance + "\n");
                    out.write("Deviation = " + Dev + "\n");
            }
            catch (IOException e)
            {
                    System.err.println("FileStream error " + e);
            }
    }

    public void updateD(double dV3, double dV2, double dV1, double
dP1, double[][] narray, int array1)
    {
            try
            {
                    out.write("\n");
                    Double ddV3 = new Double(dV3);
                    String sdV3 = ddV3.toString();
                    out.write("dV3: ");
                    out.write(sdV3);
                    out.write("    ");
                    Double ddV2 = new Double(dV2);
                    String sdV2 = ddV2.toString();
                    out.write("dV2: ");
                    out.write(sdV2);
                    out.write("    ");
                    Double ddV1 = new Double(dV1);
                    String sdV1 = ddV1.toString();
                    out.write("dV1: ");
                    out.write(sdV1);
                    out.write("    ");
                    Double ddP1 = new Double(dP1);
                    String sdP1 = ddP1.toString();
                    out.write("dP1:  ");
                    out.write(sdP1);
                    out.write("\n");

                    out.write("Node  XLoc   Yloc   Demand   Cost
MaxV   MaxP   Rmin   Price");
                    out.write("\n");
```

```java
                for (int i=0; i<arrayl; i++)
                {
                        Integer k = new Integer(i);
                        String r = k.toString();
                        out.write(r);
                        out.write("         ");

                        for (int j= 0; j<8; j++)
                        {
                                Double d = new Double(narray[i][j]);
                                String s = d.toString();
                                out.write(s);
                                out.write("      ");
                        }
                        out.write("\n");
                }
        }
        catch (IOException e)
        {
                System.err.println("FileStream error " + e);
        }
}//end updateD()


public void updateDp(double profit, double variance, double
avedev)
{
        try
        {
                Double d = new Double(profit);
                String s = d.toString();
                out.write("Avg. Profit = $" + s);
                out.write("\n");

                Double dv = new Double(variance);
                String sv = dv.toString();
                out.write("Avg. Variance = " + sv);
                out.write("\n");

                Double du = new Double(avedev);
                String su = du.toString();
                out.write("Deviation = " + su);
                out.write("\n");

        }
        catch (IOException e)
        {
                System.err.println("FileStream error " + e);
        }
}


public void close()
{
        try
```

```
                {
                        out.close();
                }
                catch (IOException e)
                {
                        System.err.println("FileStream error " + e);
                }
        }//end close()

 }//end of cDetailsOP
```

cResultsOP

```java
import java.io.*;

 public class cResultsOP
 {
        BufferedWriter out;

        public cResultsOP(double res)
        {
                try
                {
                        out = new BufferedWriter(new
FileWriter("Result2.txt"));
                        out.write("\n");
                }
                catch (IOException e)
                {
                        System.err.println("FileStream error " + e);
                }
        }//end of cResultsOP constructor


        public void updateD(double dV3, double dV2, double dV1, double
dP1, double newprofit, double CEquip, double tDemand, double
newprofit1, double newv, double unewprofit1, double unewv, double
combou)
        {
                try
                {
                        out.write("\n");
                        out.write("dV3 = " + dV3 + "  dV2 = " + dV2 + "  dP1
= " + dP1 + "  dV1 = " + dV1);
                        out.write("\n");
                        out.write("profit = " + newprofit1 + "variance = " +
newv);
                        out.write("\n");
                        out.write("Normalized Profit = " + unewprofit1);
                        out.write("\n");
                        out.write("Normalized Variance = " + unewv);
                        out.write("\n");
                        out.write("Deviation = " + combou);
                        out.write("\n");
                }
```

```
                catch (IOException e)
                {
                        System.err.println("FileStream error " + e);
                }
        }//end updateD()


        public void close()
        {
                try
                {
                        out.close();
                }
                catch (IOException e)
                {
                        System.err.println("FileStream error " + e);
                }
        }//end close()

 }//end of cResultsOP
```

ccmPV

```
/////////////ccmPV Class Changing Markets Pressure
Vessel//////////////

public class ccmPV {

        ////////Initialization/////////
        double dP1compr = 0.0;
        double dV2compr = 0.0;
        double dV1compr = 0.0;
        double dV3compr = 0.0;
        double Profitcompr = 0.0;
        double Varcompr = 0.0;
        double Dev= 0.0;

        /////////Bounds////////////
        double Tl = 6.35; //max possible thickness
        double Tu = 76.2; //min possible thickness (multiples of 6.35)
        double Lu = 7.0; //max possible length
        double Ru = 4.0; //min possible radius
        double pi = java.lang.Math.PI;

        public static void main (String[] args)
        {
                new ccmPV();
        }


        //---- Creates new clsPressureVessel ----//
```

```
      public ccmPV()
      {
            double averageprofit = 0.0;
            double averagev=1E30;
            double averageu=100000.0;//temperory variable to find best
platform
            double totalDemand2 = 0;

            ////Market space bounds and inner resolution
            double res = 0.5; //resolution of discretization
            double vor = 10.0; //origin of v-axis - x dimension
            double vlen = 20; //v-length of market
            double por = 10.0; //origin of v-axis - x dimension
            double plen = 20; //p-length of market

            double plen2 = 30;

            //call MS class constructor to create node array

            cMS MS = new cMS(res, vor, vlen, por, plen);
            double narray[][] = MS.getnarray();
            cMS MS2 = new cMS(res, vor, vlen, por, plen2);
            double narray2[][] = MS2.getnarray();

            int P1element2=0;
            int P1element=0;

            //call cDetailsOP and cResultsOP class constructor for
output
            cDetailsOP DetailsOP = new cDetailsOP(res);
            cResultsOP ResultsOP = new cResultsOP(res);

            /*Start with dP1, go through different values of dP1,
each node is assigned the max pressure according to the discretization
of P-axis. From value of max pressure, find minimum thicknesses for
each node. Then start iterating different values of dV2 which is for
finding out the radius. Then by iterating values of dV1 the length is
determined. In the end, dV3 is also iterated to find out the cutoff
point for combination. From the different values obtained, find the
profit of each node. Go on storing profits larger than previous
iteration and print and send to output file.
            */

                  double dP1; //size of first-stage pressure element
                  double P2;  //largest pressure of the considered
space

                  double P22;
                  double dP12;
                  //iterate through dP1
                  for (int i=1; i<((MS2.YDim()/2)+1) ; i++)
                  {
                  P2 = narray[i*MS.XDim()][2];
                  dP1 = P2 - narray[0][2];

                  P22 = narray2[i*MS.XDim()][2];
                  dP12 = P22 - narray2[0][2];
```

329

```
                    //check to see if dP1 is a multiple of dP2
                    int dP1factor = fCheck(plen, dP1);
                    int dP1factor2 = fCheck(plen2, dP12);

                    if ((dP1factor > 0)||(dP1factor2 > 0))
                    {;}
                    else //iterate
                    {
                    if (dP1factor <= 0)
                    {
                    int dP1node = i; //number of nodes in dP1
                    //number of elements in P direction
                    P1element = (MS.YDim()-1)/dP1node;
                    //Assign each node its max pressure
                    narray[0][6] = P2;
                    narray[0][10] = 1.0;
                    //assign nodes the pressure below Pmax line
                    double Pmax;
                    for (int a=0; a<P1element; a++)
                    {
                    int PmaxNode = MS.XDim()*dP1node +
a*dP1node*MS.XDim();
                    Pmax = narray[PmaxNode][2];
                    narray[PmaxNode][6] = Pmax;
                    narray[PmaxNode][10] = 1.0;
                    //assign nodes the pressure below Pmax line
                    int d = PmaxNode;
                    while (d > 0)
                    {
                    d = d - MS.XDim();
                    if (narray[d][10] < 1.0)
                    {
                    narray[d][6] = Pmax;
                    narray[d][10] = 1.0;
                    }
                    else {;}
                    }
                    }//end of Pmax loop
                    //assign nodes on horizontal line with Pmax
                    int b = 0;
                    while (b < (MS.NodeDim() - MS.XDim())+1)
                    {
                    for(int e=1; e<MS.XDim() ; e++)
                    {
                    narray[b+e][6] = narray[b][6];
                    narray[b+e][10] = 1.0;
                    }
                    b = b + MS.XDim();
                    }

                    }
                    else{;}


                    if (dP1factor2 <= 0)
                    {
```

```
                int dP1node2 = i; //number of nodes in dP1
                //number of elements in P direction
                P1element2 = (MS2.YDim()-1)/dP1node2;
                //Assign each node its max pressure
                narray2[0][6] = P22;
                narray2[0][10] = 1.0;
                double Pmax2;
                for (int a=0; a<P1element2; a++)
                {
                int PmaxNode2 = MS2.XDim()*dP1node2 +
a*dP1node2*MS2.XDim();
                    Pmax2 = narray2[PmaxNode2][2];
                    narray2[PmaxNode2][6] = Pmax2;
                    narray2[PmaxNode2][10] = 1.0;
                    //assign nodes the pressure below Pmax2 line
                    int d2 = PmaxNode2;
                    while (d2 > 0)
                    {
                    d2 = d2 - MS2.XDim();
                    if (narray2[d2][10] < 1.0)
                    {
                    narray2[d2][6] = Pmax2;
                    narray2[d2][10] = 1.0;
                    }
                    else {;}
                    }
                }//end of Pmax loop
                //assign nodes on horizontal line with Pmax
                int b2 = 0;
                while (b2 < (MS2.NodeDim() - MS2.XDim())+1)
                {
                for(int e=1; e<MS2.XDim() ; e++)
                {
                narray2[b2+e][6] = narray2[b2][6];
                narray2[b2+e][10] = 1.0;
                }
                b2 = b2 + MS2.XDim();
                }

                }
                else{;}

            double dV2;
            double V2h;


            for (int h=1; h<((MS.XDim()/2)+1); h++)
            {
                V2h = narray[h][1];
                dV2 = V2h - narray[0][1];
                int dVnodeh = h; //number of nodes in dV1

                //check to see if dV is a multiple of x dimension
                int Vflagh = fCheck(vlen, dV2);
                if (Vflagh > 0)
                {;}
```

```
                              else //if it is a multiple start iterating
                              {
                              //number of elements in V direction
                              int Velementh = (MS.XDim()-1)/dVnodeh;

                              //assign each node its maximum Volume
                              double Vmaxh;
                              double Rminh;
                              for (int m=1; m<Velementh+1; m++)
                              {
                              Vmaxh = narray[m*dVnodeh][1];
                              Rminh = Rmin(Vmaxh);
                              //assign nodes with max volume
                              for (int n=0; n<((m*dVnodeh)+1); n++)
                              {
                              if(narray[n][9]<1.0)
                              {
                              narray[n][7] = Rminh;
                              narray[n][9] = 1.0;
                              }
                              else {;}
                              //assign nodes above 1st line
                              if(narray2[n][9]<1.0)
                              {
                              narray2[n][7] = Rminh;
                              narray2[n][9] = 1.0;
                              }
                              else {;}

                              //assign nodes above 1st line
                              for(int p=1; p<MS.YDim(); p++)
                              {
                              if(narray[n+p*MS.XDim()][9]<1.0)
                              {
                              narray[n+p*MS.XDim()][7] = Rminh;
                              narray[n+p*MS.XDim()][9] = 1.0;
                              }
                              else{;}
                              }


                              for(int p2=1; p2<MS2.YDim(); p2++)
                              {
                              if(narray2[n+p2*MS2.XDim()][9]<1.0)
                              {
                              narray2[n+p2*MS2.XDim()][5] = Vmaxh;
                              narray2[n+p2*MS2.XDim()][7] = Rminh;
                              narray2[n+p2*MS2.XDim()][9] = 1.0;
                              }
                              else{;}
                              }

                              }
                              }//end of Vmax loop

//////////////////////////////////////////////////////////////////////
//////////
```

```
MS.initVflag();

double dV1; //size of first-stage volume element
double V2s; //largest volume of the considered space

MS2.initVflag();


//go through dV1s
for (int j=1; j<((MS.XDim()/2)+1); j++)
{
     V2s = narray[j][1];
     dV1 = V2s - narray[0][1];
     int dVnode = j; //number of nodes in dV1

     //check to see if dV is a multiple of the v-dimension
     int Vflag = fCheck(dV2, dV1);
     if (Vflag > 0)
     {;}
     else //if it is a multiple iterate
     {
     //number of elements in V direction
     int Velement = (MS.XDim()-1)/dVnode;

     ///assign each node its max Volume
     double Vmax;
     double Rmin;
     for (int z=1; z<Velement+1; z++)
     {
     Vmax = narray[z*dVnode][1];
     Rmin = Rmin(Vmax);
     //assign nodes with Vmax
     for (int y=0; y<((z*dVnode)+1); y++)
     {
     if(narray[y][9]<1.0)
     {
     narray[y][5] = Vmax;
     narray[y][9] = 1.0;
     }
     else {;}
     //assign nodes above v axis line
     if(narray2[y][9]<1.0)
     {
     narray2[y][5] = Vmax;
     narray2[y][9] = 1.0;
     }
     else {;}


     for(int x=1; x<MS.YDim(); x++)
     {
     if(narray[y+x*MS.XDim()][9]<1.0)
     {
     narray[y+x*MS.XDim()][5] = Vmax;
     narray[y+x*MS.XDim()][9] = 1.0;
```

```
        }
        else{;}
        }

        for(int x2=1; x2<MS2.YDim(); x2++)
        {
        if(narray2[y+x2*MS2.XDim()][9]<1.0)
        {
        narray2[y+x2*MS.XDim()][5] = Vmax;
        narray2[y+x2*MS.XDim()][9] = 1.0;
        }
        else{;}
        }

        }
        }//end of Vmax loop

        double de2[];
        double pr2[];
        de2=new double[5];
        pr2=new double[5];
        double v2=0.0;
        double unewv2=0.0;
        double unewprofit12=0.0;
        double combou1=0.0;
        double dunewv2=0.0;
        double dunewprofit12=0.0;

        double CEquip2 = 0;
        double newprofit12=0;

        double newprofit1=0;
        double newv22 = 0;
        double newv2f = 0;
        double Locost2 = 0;


        totalDemand2 = 0;
        double newprofit2 = 0;
        /*Start iterating through dV3. Assume cutoff point
and recalculate
        profit considering that many vessels. Find the best
combination of
        different parameters
        */


        double dV3; //size of third space element
        double V3; //cutoff point
        //first try when there is no dV3 - no cutoff
        dV3 = vlen;
        //calculate cost for each node
        for (int n=0; n<MS.NodeDim(); n++)
        {
        double V = narray[n][1];
        double R = narray[n][7];
        double maxV = narray[n][5];
```

334

```
double maxP = narray[n][6];
double Lo = (maxV / (pi*R*R)) - ((4.0/3.0)*R);
double L = (V / (pi*R*R)) - ((4.0/3.0)*R);
double cost = Cost(maxP, V, R, L, Lo);
narray[n][4] = cost;


}


//find avg profit of entire space
double newprofit = 0;
double totalDemand = 0;


double de[];
double pr[];
double newv=0.0;
de=new double[5];
pr=new double[5];
double v=0.0;
double unewv=0.0;
double unewprofit1=0.0;
double combou=0.0;
double dunewv=0.0;
double dunewprofit1=0.0;

for (int q = 0; q<MS.NodeDim(); q++)
{
//demand * (selling price - cost)

de[0]=narray[q][3];
de[1]=0.8*narray[q][3];
de[2]=0.9*narray[q][3];
de[3]=1.1*narray[q][3];
de[4]=1.2*narray[q][3];

pr[0]=de[0]*(narray[q][8] - narray[q][4]);


for(int g=1; g<5; g++)
{
    pr[g]=de[g]*(narray[q][8] - narray[q][4]);
    v=v+((pr[g]-pr[0])*(pr[g]-pr[0]))/3;

}

newv=newv+v;


newprofit = newprofit + (narray[q][3]*(narray[q][8] -
narray[q][4]));

totalDemand = narray[q][3] + totalDemand;
}
// CEquip for each dV1
double CEquip = 0;
for (int p=1; p<(Velementh+1); p++)
{
```

```
double rad = narray[p*dVnodeh][7];
CEquip = CEquip + (500000 + (50000*rad));
}

//Cost of each Lo and each dP1
double Locost = 250*Velement+100*P1element;
newprofit1 = newprofit - CEquip - Locost;

//divide by total demand of space to take average

unewprofit1=newprofit1/1.7E9;


unewv=1E16/newv;

if((unewprofit1<0)||(unewprofit1>1))
      unewprofit1=0;

if((unewv<0)||(unewv>1))
      unewv=0;

dunewprofit1=1-unewprofit1;
dunewv=1-unewv;


dV3 = vlen;
//calculate cost for each node
for (int n=0; n<MS2.NodeDim(); n++)
{
double V2 = narray2[n][1];
double R2 = narray2[n][7];
double maxV2 = narray2[n][5];
double maxP2 = narray2[n][6];
double Lo2 = (maxV2 / (pi*R2*R2)) - ((4.0/3.0)*R2);
double L2 = (V2 / (pi*R2*R2)) - ((4.0/3.0)*R2);
double cost2 = Cost(maxP2, V2, R2, L2, Lo2);
narray2[n][4] = cost2;
}
//find avg profit of entire space
double newv2 = 0;
v2=0.0;
for (int q = 0; q<MS2.NodeDim(); q++)
{
//demand * (selling price - cost)

if (q>1680)
      narray2[q][3] = 2+narray2[q][3];

de2[0]=narray2[q][3];
de2[1]=0.8*narray2[q][3];
de2[2]=0.9*narray2[q][3];
de2[3]=1.1*narray2[q][3];
de2[4]=1.2*narray2[q][3];

pr2[0]=de2[0]*(narray2[q][8] - narray2[q][4]);
```

```
                    for(int g=1; g<5; g++)
                    {
                            pr2[g]=de2[g]*(narray2[q][8] - narray2[q][4]);
                            v2=v2+((pr2[g]-pr2[0])*(pr2[g]-pr2[0]))/3;

                    }

                    newv2=newv2+v2;

                    newprofit2 = newprofit2 +
(narray2[q][3]*(narray2[q][8] - narray2[q][4]));
                    totalDemand2 = narray2[q][3] + totalDemand2;
                    }


                    for (int p=1; p<(Velementh+1); p++)
                    {
                    double rad2 = narray2[p*dVnodeh][7];
                    CEquip2 = CEquip2 + (500000 + (50000*rad2));
                    }

                    Locost2 = 250*Velement+100*P1element2;
                    newprofit12 = newprofit2 - CEquip2 - Locost2;

                    unewprofit12=newprofit12/1E10;


                    unewv2=1E16/newv2;



                    dunewprofit12=1-unewprofit12;
                    dunewv2=1-unewv2;


                    combou1=0.5*dunewv2+0.5*dunewprofit12;


                    combou=0.008*dunewv+0.012*dunewprofit1+0.98*combou1;
//                  combou=0.5*dunewv+0.5*dunewprofit1+0*combou1;


                    if (combou < averageu)
                    {
                    averageu = combou;
                    dP1compr = dP1;
                    dV2compr = dV2;
                    dV1compr = dV1;
                    dV3compr = vlen;
                    Profitcompr = newprofit1;
                    Varcompr = newv;
                    Dev= averageu;
                    //System.out.println("dP1"+dP1+" dV2"+dV2+"
dV1"+dV1+" dV3"+dV3+" combou"+combou);
```

```
                    DetailsOP.updateD(dV3, dV2, dV1, dP1, narray,
MS.NodeDim());
                    DetailsOP.updateDp(newprofit1, newv, averageu);
                    }

                    ResultsOP.updateD(dV3, dV2, dV1, dP1, newprofit,
CEquip, totalDemand, newprofit1, newv, dunewprofit1, dunewv, combou);

                    //iterate through dV3s
                    for (int c=((MS.XDim()/2)+1); c<MS.XDim(); c++)
                    {
                    int dV3node = c;
                    V3 = narray[dV3node-1][1];
                    dV3 = V3 - narray[0][1];

                    //check to see if dV3 is a multiple of dV1
                    int dV3flag = fCheck(dV3, dV2);
                    if (dV3flag > 0)
                    {;}
                    else //if it is a multiple iterate
                    {

                    //calculate cost for each node
                    //need R, Pmax, Lo, and V(node)
                    for (int n=0; n<MS.NodeDim(); n++)
                    {
                    double V = narray[n][1];
                    double R = narray[n][7];
                    double maxV = narray[n][5];
                    double maxP = narray[n][6];
                    double Lo = (maxV / (pi*R*R)) - ((4.0/3.0)*R);
                    double L = (V / (pi*R*R)) - ((4.0/3.0)*R);
                    double cost = Cost(maxP, V, R, L, Lo);
                    narray[n][4] = cost;

                    }


                    for (int n=0; n<MS2.NodeDim(); n++)
                    {
                    double V2 = narray2[n][1];
                    double R2 = narray2[n][7];
                    double maxV2 = narray2[n][5];
                    double maxP2 = narray2[n][6];
                    double Lo2 = (maxV2 / (pi*R2*R2)) - ((4.0/3.0)*R2);
                    double L2 = (V2 / (pi*R2*R2)) - ((4.0/3.0)*R2);
                    double cost2 = Cost(maxP2, V2, R2, L2, Lo2);
                    narray2[n][4] = cost2;
                    }


                    for (int d = dV3node; d<MS.XDim(); d++)
                    {
                    for (int e=0; e<MS.YDim(); e++)
                    {
                    int node = d/2 + (e*MS.XDim());
                    //location of half of current node
```

```
narray[(d+(e*MS.XDim()))][4] = 2 * narray[node][4];
}
}

//find avg profit

newprofit = 0;
totalDemand = 0;
v=0;
newv=0;
for (int q = 0; q<MS.NodeDim(); q++)
{

de[0]=narray[q][3];
de[1]=0.8*narray[q][3];
de[2]=0.9*narray[q][3];
de[3]=1.1*narray[q][3];
de[4]=1.2*narray[q][3];

pr[0]=de[0]*(narray[q][8] - narray[q][4]);


for(int g=1; g<5; g++)
{
        pr[g]=de[g]*(narray[q][8] - narray[q][4]);
        v=v+((pr[g]-pr[0])*(pr[g]-pr[0]))/3;
}

newv=newv+v;

//demand * (selling price - cost)
newprofit = newprofit + (narray[q][3]*(narray[q][8] -
narray[q][4]));
totalDemand = narray[q][3] + totalDemand;
}

newprofit2 = 0;
totalDemand2 = 0;

newv2=0;
newv2f=0;

v2=0.0;
for (int q = 0; q<MS2.NodeDim(); q++)
{
//demand * (selling price - cost)

if (q>1680)
      narray2[q][3] = 2+narray2[q][3];

de2[0]=narray2[q][3];
de2[1]=0.8*narray2[q][3];
de2[2]=0.9*narray2[q][3];
de2[3]=1.1*narray2[q][3];
de2[4]=1.2*narray2[q][3];

pr2[0]=de2[0]*(narray2[q][8] - narray2[q][4]);
```

```
                        for(int g=1; g<5; g++)
                        {
                                pr2[g]=de2[g]*(narray2[q][8] - narray2[q][4]);
                                v2=v2+((pr2[g]-pr2[0])*(pr2[g]-pr2[0]))/3;

                        }

                        newv2=newv2+v2;
                        newprofit2 = newprofit2 +
(narray2[q][3]*(narray2[q][8] - narray2[q][4]));
                        totalDemand2 = narray2[q][3] + totalDemand2;
                        }


                        CEquip = 0;
                        int o = 1;
                        int onode = dVnode;
                        while (onode < dV3node)
                        {
                        double rad = narray[onode][7];
                        CEquip = CEquip + (500000 + (50000*rad));
                        o++;
                        onode = o*dVnode;
                        }

                        newprofit1 = newprofit - CEquip - Locost;

                        unewprofit1 = newprofit1/1.7E9;

                        newprofit12 = newprofit2 - CEquip - Locost2;

//                      newprofit12 = newprofit12 / totalDemand2;

                        unewprofit12 = newprofit12/5E9;

                        unewv = 1E16/newv;
                        unewv2 = 1E16/newv2;


                        if((unewprofit1<0)||(unewprofit1>1))
                              unewprofit1=0;

                        if((unewprofit12<0)||(unewprofit12>1))
                              unewprofit12=0;

                        if((unewv<0)||(unewv>1))
                              unewv=0;

                        dunewprofit1=1-unewprofit1;
                        dunewv=1-unewv;


                        if((unewv2<0)||(unewv2>1))
                              unewv2=0;
```

```
                dunewprofit12=1-unewprofit12;
                dunewv2=1-unewv2;


                combou1=0.5*dunewv2+0.5*dunewprofit12;

                combou=0.008*dunewv+0.012*dunewprofit1+0.98*combou1;
    //          combou=0.5*dunewv+0.5*dunewprofit1+0*combou1;

                if (combou < averageu)
                {
                averageu = combou;
                dP1compr = dP1;
                dV2compr = dV2;
                dV1compr = dV1;
                dV3compr = vlen;
                Profitcompr = newprofit12;
                Varcompr = newv;
                Dev= averageu;
                //System.out.println("dP1"+dP1+" dV2"+dV2+"
dV1"+dV1+" dV3"+dV3+" combou"+combou);

                //write results to file
                DetailsOP.updateD(dV3, dV2, dV1, dP1, narray,
MS.NodeDim());
                DetailsOP.updateDp(newprofit1, newv, combou);
                }

                ResultsOP.updateD(dV3, dV2, dV1, dP1, newprofit,
CEquip, totalDemand, newprofit1, newv, dunewprofit1, dunewv, combou);

                MS.initCost();

                MS2.initCost();
                }//end of else in dV3 iteration

                }//end of dV3 iteration


                //initialize flags for next iteration

                MS.initVmax();
                MS.initVflag();

                MS2.initVmax();
                MS2.initVflag();

            } //end of else of dV1
            }//end of dV1


            MS.initVflag();

            MS2.initVflag();

            }//end of else of dV2
            } //end of dV2
```

341

```java
            MS.initPmax();
            MS.initPflag();

            MS2.initPmax();
            MS2.initPflag();

            }//end of else of dP1
            }//end of dP1

            //final results
            DetailsOP.updateR(dV3compr, dV2compr, dV1compr, dP1compr,
Profitcompr, Varcompr, Dev);

            //close output files
            DetailsOP.close();
            ResultsOP.close();

        } //end of ccmPV() constructor


        //fCheck()//
        public int fCheck(double n, double m)
        {
            Double x = new Double(n/m);
            int y = x.intValue();
            double z = y+0.0;
            double fractioncheck = (n/m) - z;
            //System.out.println("fractioncheck = " + fractioncheck);
            int flag = 0;
            if (fractioncheck < 0 || fractioncheck > 0)
            {
                    flag = 1;
            }
            return flag;
        }//end of fCheck()

        ///Cost()//
        public double Cost(double xP, double xV, double xR, double xL,
double xLo)
        {
            double P = xP; //Pressure, MPa
            double V = xV; //Volume, m^3
            double R = xR; //Radius, m
            double L = xL; //Length, m
            double Lo = xLo; //Raw Length, m

            //---- Constants ----//
            double sigy = 1077; //Yield strength, MPa
            double dens = 7800; //density, kg/m^3
            double Cs = 0.8; //processed shell stell cost, $/kg
            double Ch = 2.0; //cost of forged steel for head, $/kg
            double Cp = 0.3; //cost of raw steel plate, $/kg
            double Cw = 15; //cost of welding material, $/kg

            double Ts = (R*1000)* (P / (sigy - (0.6*P))); //shell
thickness, mm
```

```java
            double Th = (R*1000)* (P / ( (2*sigy) - (0.2*P) )); //head
thickness, mm

            double CWaste = 2 * pi * dens * Cp * (Ts/1000) * R * (Lo -
L); //cost of wasted material, $
            double CMat = 2 * pi * dens * ( (Cs*R*(Ts/1000)*L) +
(Ch*R*R*(Th/1000)) ) + CWaste; //material cost, $
            double VLWeld = (4.0/9.0) * pi * (Ts/1000) * (Ts/1000) * L;
//volume of long weld, m^3
            double CLongWeld = VLWeld * dens * Cw; //cost of long weld
            double VCWeld = (8.0/9.0) * pi * pi * (Ts/1000) * (Ts/1000)
* R; //volume of circ. weld, m^3
            double CCircWeld = VCWeld * dens * Cw; //cost of circular
weld, $
            double CWeld = CLongWeld + CCircWeld; //total cost of weld

            double CTotal = CMat + CWeld; //total cost excluding
equipment cost, $
            return CTotal;
        } //end of Cost


        public double Rmin(double xV)
        {
            //find the minimum feasible radius for a vessel
            //with given Volume
            double V = xV;
            double Verr = 0.0; //used to compare with given V
            double R1 = 0.0;
            double R2 = Ru;
            double Rmin = 0.0; //minimum radius
            double error = 1.0; //error used for testing convergence
            //iterate through volume quation with bisection method
            for(int i=0; i<30; i++)
            {
                Rmin = (R1+R2)/2;
                Verr = (4/3)*pi*(Rmin*Rmin*Rmin) + pi*Rmin*Rmin*Lu;
                error = java.lang.Math.abs(Verr - V);
                if (error < 0.01) { break; }
                if (Verr < V)
                {
                    R1 = Rmin;
                }
                else
                {
                    R2 = Rmin;
                }
            }
            return Rmin;
        }//end of Rmin

} //end of ccmPV
```

APPENDIX B

PRODUCT PLATFORM SPECIFICATIONS OF THE PRESSURE VESSEL EXAMPLE

The results of the product platform for the different scenarios are discussed in Section 4.1.4 and Section 5.3. A section of the platform specifications are presented in these sections. The complete mapping of the product platform specifications in the space of customization is presented in this appendix. In the first part, the specifications of the scenarios of Chapter 4 having uncertain distribution of demand is listed while in the latter the specifications of the scenarios of Chapter 5 having changing extent of market spaces is listed.

Chapter 4 Pressure Vessel Scenarios

A section of the results are presented in Table 4.2, Table 4.4 and Table 4.5.

Scenario 1 - Certain Distribution of Demand

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
15	10	0.783	7.311	3.638	6.744
	10.5		7.679	3.821	
	11		8.047	4.003	
	11.5		8.415	4.185	
	12		8.783	4.367	
	12.5		9.151	4.549	
	13		9.520	4.731	
	13.5		9.889	4.914	
	14		10.258	5.096	
	14.5		10.628	5.278	
	15		10.997	5.460	
	15.5		11.367	3.638	
	16		11.737	3.821	
	16.5		12.107	4.003	

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
	17		12.478	4.185	
	17.5		12.848	4.367	
	18		13.219	4.549	
	18.5		13.590	4.731	
	19		13.961	4.914	
	19.5		14.333	5.096	
	20		14.704	7.284	
	20.5		15.076	5.460	
	21		15.448	5.643	
	21.5		15.820	5.825	
	22		16.193	6.007	
	22.5		16.566	6.189	
	23		16.938	6.372	
	23.5		17.312	6.554	
	24		17.685	6.737	
	24.5		18.058	6.919	
	25		18.432	9.109	
	25.5		18.806	7.284	
	26		19.180	7.466	
	26.5		19.555	7.649	
	27		19.929	7.831	
	27.5		20.304	8.014	
	28		20.679	8.196	
	28.5		21.054	8.379	
	29		21.430	10.570	
	29.5		21.805	8.744	
	30		22.181	10.936	
20	10	0.898	8.385	4.173	6.697
	10.5		8.806	4.382	
	11		9.228	4.591	
	11.5		9.650	4.799	
	12		10.073	5.008	
	12.5		10.496	5.217	
	13		10.918	5.426	
	13.5		11.342	5.635	
	14		11.765	5.844	
	14.5		12.189	6.053	
	15		12.612	6.262	
	15.5		13.036	6.471	
	16		13.461	6.680	
	16.5		13.885	6.889	
	17		14.310	7.098	
	17.5		14.735	7.308	
	18		15.160	7.517	
	18.5		15.586	7.726	
	19		16.012	7.935	

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
	19.5		16.438	8.144	
	20		16.864	8.353	
	20.5		17.290	8.563	
	21		17.717	8.772	
	21.5		18.144	8.981	
	22		18.571	9.191	
	22.5		18.999	9.400	
	23		19.426	9.609	
	23.5		19.854	9.819	
	24		20.282	10.028	
	24.5		20.711	10.237	
	25		21.139	10.447	
	25.5		21.568	10.656	
	26		21.997	10.866	
	26.5		22.427	11.075	
	27		22.856	11.285	
	27.5		23.286	11.494	
	28		23.716	11.704	
	28.5		24.147	11.913	
	29		24.577	12.123	
	29.5		25.008	12.332	
	30		25.439	12.542	
25	10	0.998	9.318	4.638	6.659
	10.5		9.787	4.870	
	11		10.256	5.102	
	11.5		10.725	5.334	
	12		11.195	5.566	
	12.5		11.664	5.798	
	13		12.134	6.030	
	13.5		12.605	6.263	
	14		13.075	6.495	
	14.5		13.546	6.727	
	15		14.017	6.960	
	15.5		14.488	7.192	
	16		14.960	7.424	
	16.5		15.432	7.657	
	17		15.904	7.889	
	17.5		16.376	8.121	
	18		16.849	8.354	
	18.5		17.322	8.586	
	19		17.795	8.819	
	19.5		18.268	9.051	
	20		18.742	9.284	
	20.5		19.216	9.516	
	21		19.690	9.749	
	21.5		20.164	9.981	
	22		20.639	10.214	

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
	22.5		21.114	10.447	
	23		21.590	10.679	
	23.5		22.065	10.912	
	24		22.541	11.145	
	24.5		23.017	11.377	
	25		23.493	11.610	
	25.5		23.970	11.843	
	26		24.447	12.076	
	26.5		24.924	12.308	
	27		25.402	12.541	
	27.5		25.879	12.774	
	28		26.357	13.007	
	28.5		26.836	13.240	
	29		27.314	13.473	
	29.5		27.793	13.706	
	30		28.272	13.939	
30	10	1.087	7.313	3.639	14.523
	10.5		7.681	3.822	
	11		8.049	4.004	
	11.5		8.417	4.186	
	12		8.785	4.368	
	12.5		9.154	4.550	
	13		9.523	4.733	
	13.5		9.892	4.915	
	14		10.261	5.097	
	14.5		10.630	5.279	
	15		11.000	5.462	
	15.5		14.488	7.192	
	16		14.960	7.424	
	16.5		15.432	7.657	
	17		15.904	7.889	
	17.5		16.376	8.121	
	18		16.849	8.354	
	18.5		17.322	8.586	
	19		17.795	8.819	
	19.5		18.268	9.051	
	20		18.742	9.284	
	20.5		19.216	9.516	
	21		19.690	9.749	
	21.5		20.164	9.981	
	22		20.639	10.214	
	22.5		21.114	10.447	
	23		21.590	10.679	
	23.5		22.065	10.912	
	24		22.541	11.145	
	24.5		23.017	11.377	
	25		23.493	11.610	

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
	25.5		23.970	11.843	
	26		24.447	12.076	
	26.5		24.924	12.308	
	27		25.402	12.541	
	27.5		25.879	12.774	
	28		26.357	13.007	
	28.5		26.836	13.240	
	29		27.314	13.473	
	29.5		27.793	13.706	
	30		28.272	13.939	

Scenario 2 – Uncertain Distribution of Demand

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
10	10	0.718	6.701	3.335	5.223
	12		8.050	4.003	
	14		9.403	4.671	
	16		10.758	5.339	
	18		12.116	6.008	
	20		13.478	6.676	
	22		14.842	7.345	
	24		16.210	8.015	
	26		17.581	8.684	
	28		18.955	9.354	
	30		20.331	10.024	
10.5	10		6.701	3.335	5.532
	12		8.050	4.003	
	14		9.403	4.671	
	16		10.758	5.339	
	18		12.116	6.008	
	20		13.478	6.676	
	22		14.842	7.345	
	24		16.210	8.015	
	26		17.581	8.684	
	28		18.955	9.354	
	30		20.331	10.024	
11	10		6.701	3.335	5.841
	12		8.050	4.003	
	14		9.403	4.671	
	16		10.758	5.339	
	18		12.116	6.008	
	20		13.478	6.676	
	22		14.842	7.345	
	24		16.210	8.015	
	26		17.581	8.684	
	28		18.955	9.354	

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
	30		20.331	10.024	
11.5	10		6.701	3.335	6.150
	12		8.050	4.003	
	14		9.403	4.671	
	16		10.758	5.339	
	18		12.116	6.008	
	20		13.478	6.676	
	22		14.842	7.345	
	24		16.210	8.015	
	26		17.581	8.684	
	28		18.955	9.354	
	30		20.331	10.024	
12	10		6.701	3.335	6.459
	12		8.050	4.003	
	14		9.403	4.671	
	16		10.758	5.339	
	18		12.116	6.008	
	20		13.478	6.676	
	22		14.842	7.345	
	24		16.210	8.015	
	26		17.581	8.684	
	28		18.955	9.354	
	30		20.331	10.024	
12.5	10		6.701	3.335	6.768
	12		8.050	4.003	
	14		9.403	4.671	
	16		10.758	5.339	
	18		12.116	6.008	
	20		13.478	6.676	
	22		14.842	7.345	
	24		16.210	8.015	
	26		17.581	8.684	
	28		18.955	9.354	
	30		20.331	10.024	
13	10	0.783	7.311	3.638	5.705
	12		8.783	4.367	
	14		10.258	5.096	
	16		11.737	5.825	
	18		13.219	6.554	
	20		14.704	7.284	
	22		16.193	8.014	
	24		17.685	8.744	
	26		19.180	9.474	
	28		20.679	10.205	
	30		22.181	10.936	
13.5	10		7.311	3.638	5.965

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
	12		8.783	4.367	
	14		10.258	5.096	
	16		11.737	5.825	
	18		13.219	6.554	
	20		14.704	7.284	
	22		16.193	8.014	
	24		17.685	8.744	
	26		19.180	9.474	
	28		20.679	10.205	
	30		22.181	10.936	
14	10		7.311	3.638	6.225
	12		8.783	4.367	
	14		10.258	5.096	
	16		11.737	5.825	
	18		13.219	6.554	
	20		14.704	7.284	
	22		16.193	8.014	
	24		17.685	8.744	
	26		19.180	9.474	
	28		20.679	10.205	
	30		22.181	10.936	
14.5	10		7.311	3.638	6.484
	12		8.783	4.367	
	14		10.258	5.096	
	16		11.737	5.825	
	18		13.219	6.554	
	20		14.704	7.284	
	22		16.193	8.014	
	24		17.685	8.744	
	26		19.180	9.474	
	28		20.679	10.205	
	30		22.181	10.936	
15	10		7.311	3.638	6.744
	12		8.783	4.367	
	14		10.258	5.096	
	16		11.737	5.825	
	18		13.219	6.554	
	20		14.704	7.284	
	22		16.193	8.014	
	24		17.685	8.744	
	26		19.180	9.474	
	28		20.679	10.205	
	30		22.181	10.936	
15.5	10	0.843	7.868	3.916	5.824
	12		9.453	4.700	
	14		11.040	5.484	
	16		12.632	6.269	

350

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
	18		14.227	7.054	
	20		15.825	7.839	
	22		17.428	8.625	
	24		19.033	9.410	
	26		20.643	10.196	
	28		22.256	10.983	
	30		23.873	11.770	
16	10		7.868	3.916	6.048
	12		9.453	4.700	
	14		11.040	5.484	
	16		12.632	6.269	
	18		14.227	7.054	
	20		15.825	7.839	
	22		17.428	8.625	
	24		19.033	9.410	
	26		20.643	10.196	
	28		22.256	10.983	
	30		23.873	11.770	
16.5	10		7.868	3.916	6.272
	12		9.453	4.700	
	14		11.040	5.484	
	16		12.632	6.269	
	18		14.227	7.054	
	20		15.825	7.839	
	22		17.428	8.625	
	24		19.033	9.410	
	26		20.643	10.196	
	28		22.256	10.983	
	30		23.873	11.770	
17	10		7.868	3.916	6.496
	12		9.453	4.700	
	14		11.040	5.484	
	16		12.632	6.269	
	18		14.227	7.054	
	20		15.825	7.839	
	22		17.428	8.625	
	24		19.033	9.410	
	26		20.643	10.196	
	28		22.256	10.983	
	30		23.873	11.770	
17.5	10		7.868	3.916	6.720
	12		9.453	4.700	
	14		11.040	5.484	
	16		12.632	6.269	
	18		14.227	7.054	
	20		15.825	7.839	
	22		17.428	8.625	

351

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
	24		19.033	9.410	
	26		20.643	10.196	
	28		22.256	10.983	
	30		23.873	11.770	
18	10	0.898	8.384	4.172	5.909
	12		10.072	5.008	
	14		11.764	5.844	
	16		13.459	6.680	
	18		15.159	7.516	
	20		16.862	8.353	
	22		18.569	9.190	
	24		20.280	10.027	
	26		21.995	10.864	
	28		23.714	11.702	
	30		25.436	12.541	
18.5	10		8.384	4.172	6.107
	12		10.072	5.008	
	14		11.764	5.844	
	16		13.459	6.680	
	18		15.159	7.516	
	20		16.862	8.353	
	22		18.569	9.190	
	24		20.280	10.027	
	26		21.995	10.864	
	28		23.714	11.702	
	30		25.436	12.541	
19	10		8.384	4.172	6.304
	12		10.072	5.008	
	14		11.764	5.844	
	16		13.459	6.680	
	18		15.159	7.516	
	20		16.862	8.353	
	22		18.569	9.190	
	24		20.280	10.027	
	26		21.995	10.864	
	28		23.714	11.702	
	30		25.436	12.541	
19.5	10		8.384	4.172	6.502
	12		10.072	5.008	
	14		11.764	5.844	
	16		13.459	6.680	
	18		15.159	7.516	
	20		16.862	8.353	
	22		18.569	9.190	
	24		20.280	10.027	
	26		21.995	10.864	
	28		23.714	11.702	

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
	30		25.436	12.541	
20	10		8.384	4.172	6.699
	12		10.072	5.008	
	14		11.764	5.844	
	16		13.459	6.680	
	18		15.159	7.516	
	20		16.862	8.353	
	22		18.569	9.190	
	24		20.280	10.027	
	26		21.995	10.864	
	28		23.714	11.702	
	30		25.436	12.541	
20.5	10	0.949	8.861	4.410	5.980
	12		10.645	5.293	
	14		12.433	6.176	
	16		14.225	7.060	
	18		16.021	7.944	
	20		17.822	8.828	
	22		19.626	9.713	
	24		21.434	10.597	
	26		23.247	11.483	
	28		25.063	12.368	
	30		26.884	13.254	
21	10		8.861	4.410	6.157
	12		10.645	5.293	
	14		12.433	6.176	
	16		14.225	7.060	
	18		16.021	7.944	
	20		17.822	8.828	
	22		19.626	9.713	
	24		21.434	10.597	
	26		23.247	11.483	
	28		25.063	12.368	
	30		26.884	13.254	
21.5	10		8.861	4.410	6.334
	12		10.645	5.293	
	14		12.433	6.176	
	16		14.225	7.060	
	18		16.021	7.944	
	20		17.822	8.828	
	22		19.626	9.713	
	24		21.434	10.597	
	26		23.247	11.483	
	28		25.063	12.368	
	30		26.884	13.254	
22	10		8.861	4.410	6.510
	12		10.645	5.293	

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
	14		12.433	6.176	
	16		14.225	7.060	
	18		16.021	7.944	
	20		17.822	8.828	
	22		19.626	9.713	
	24		21.434	10.597	
	26		23.247	11.483	
	28		25.063	12.368	
	30		26.884	13.254	
22.5	10		8.861	4.410	6.687
	12		10.645	5.293	
	14		12.433	6.176	
	16		14.225	7.060	
	18		16.021	7.944	
	20		17.822	8.828	
	22		19.626	9.713	
	24		21.434	10.597	
	26		23.247	11.483	
	28		25.063	12.368	
	30		26.884	13.254	
23	10	0.997	9.309	4.633	6.036
	12		11.183	5.561	
	14		13.062	6.488	
	16		14.945	7.417	
	18		16.832	8.345	
	20		18.723	9.274	
	22		20.619	10.204	
	24		22.518	11.133	
	26		24.422	12.063	
	28		26.331	12.994	
	30		28.244	13.925	
23.5	10		9.309	4.633	6.196
	12		11.183	5.561	
	14		13.062	6.488	
	16		14.945	7.417	
	18		16.832	8.345	
	20		18.723	9.274	
	22		20.619	10.204	
	24		22.518	11.133	
	26		24.422	12.063	
	28		26.331	12.994	
	30		28.244	13.925	
24	10		9.309	4.633	6.356
	12		11.183	5.561	
	14		13.062	6.488	
	16		14.945	7.417	
	18		16.832	8.345	

354

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
	20		18.723	9.274	
	22		20.619	10.204	
	24		22.518	11.133	
	26		24.422	12.063	
	28		26.331	12.994	
	30		28.244	13.925	
24.5	10		9.309	4.633	6.516
	12		11.183	5.561	
	14		13.062	6.488	
	16		14.945	7.417	
	18		16.832	8.345	
	20		18.723	9.274	
	22		20.619	10.204	
	24		22.518	11.133	
	26		24.422	12.063	
	28		26.331	12.994	
	30		28.244	13.925	
25	10		9.309	4.633	6.676
	12		11.183	5.561	
	14		13.062	6.488	
	16		14.945	7.417	
	18		16.832	8.345	
	20		18.723	9.274	
	22		20.619	10.204	
	24		22.518	11.133	
	26		24.422	12.063	
	28		26.331	12.994	
	30		28.244	13.925	
25.5	10	1.043	9.739	4.847	6.071
	12		11.699	5.817	
	14		13.665	6.788	
	16		15.634	7.759	
	18		17.608	8.730	
	20		19.587	9.702	
	22		21.570	10.675	
	24		23.557	11.647	
	26		25.549	12.620	
	28		27.546	13.593	
	30		29.547	14.567	
26	10		9.739	4.847	6.217
	12		11.699	5.817	
	14		13.665	6.788	
	16		15.634	7.759	
	18		17.608	8.730	
	20		19.587	9.702	
	22		21.570	10.675	

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
	24		23.557	11.647	
	26		25.549	12.620	
	28		27.546	13.593	
	30		29.547	14.567	
26.5	10		9.739	4.847	6.363
	12		11.699	5.817	
	14		13.665	6.788	
	16		15.634	7.759	
	18		17.608	8.730	
	20		19.587	9.702	
	22		21.570	10.675	
	24		23.557	11.647	
	26		25.549	12.620	
	28		27.546	13.593	
	30		29.547	14.567	
27	10		9.739	4.847	6.510
	12		11.699	5.817	
	14		13.665	6.788	
	16		15.634	7.759	
	18		17.608	8.730	
	20		19.587	9.702	
	22		21.570	10.675	
	24		23.557	11.647	
	26		25.549	12.620	
	28		27.546	13.593	
	30		29.547	14.567	
27.5	10		9.739	4.847	6.656
	12		11.699	5.817	
	14		13.665	6.788	
	16		15.634	7.759	
	18		17.608	8.730	
	20		19.587	9.702	
	22		21.570	10.675	
	24		23.557	11.647	
	26		25.549	12.620	
	28		27.546	13.593	
	30		29.547	14.567	
28	10	1.086	10.140	5.046	6.109
	12		12.182	6.057	
	14		14.228	7.068	
	16		16.279	8.079	
	18		18.334	9.090	
	20		20.394	10.102	
	22		22.459	11.115	
	24		24.529	12.127	
	26		26.603	13.140	

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
	28		28.681	14.154	
	30		30.765	15.168	
28.5	10		10.140	5.046	6.244
	12		12.182	6.057	
	14		14.228	7.068	
	16		16.279	8.079	
	18		18.334	9.090	
	20		20.394	10.102	
	22		22.459	11.115	
	24		24.529	12.127	
	26		26.603	13.140	
	28		28.681	14.154	
	30		30.765	15.168	
29	10		10.140	5.046	6.379
	12		12.182	6.057	
	14		14.228	7.068	
	16		16.279	8.079	
	18		18.334	9.090	
	20		20.394	10.102	
	22		22.459	11.115	
	24		24.529	12.127	
	26		26.603	13.140	
	28		28.681	14.154	
	30		30.765	15.168	
29.5	10		10.140	5.046	6.514
	12		12.182	6.057	
	14		14.228	7.068	
	16		16.279	8.079	
	18		18.334	9.090	
	20		20.394	10.102	
	22		22.459	11.115	
	24		24.529	12.127	
	26		26.603	13.140	
	28		28.681	14.154	
	30		30.765	15.168	
30	10		10.140	5.046	6.649
	12		12.182	6.057	
	14		14.228	7.068	
	16		16.279	8.079	
	18		18.334	9.090	
	20		20.394	10.102	
	22		22.459	11.115	
	24		24.529	12.127	
	26		26.603	13.140	
	28		28.681	14.154	
	30		30.765	15.168	

357

Chapter 5

A section of the results are presented in Table 5.1, Table 5.3 and Table 5.4.

Scenario 1 - Design considering present markets

(V 10-15m^3)

V (m^3)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
10	10	0.718	6.701	3.335	5.223
	10.5		7.038	3.502	
	11		7.375	3.669	
	11.5		7.713	3.836	
	12		8.050	4.003	
	12.5		8.388	4.170	
	13		8.726	4.337	
	13.5		9.064	4.504	
	14		9.403	4.671	
	14.5		9.741	4.838	
	15		10.080	5.005	
	15.5		10.419	5.172	
	16		10.758	5.339	
	16.5		11.097	5.506	
	17		11.437	5.673	
	17.5		11.777	5.840	
	18		12.116	6.008	
	18.5		12.457	6.175	
	19		12.797	6.342	
	19.5		13.137	6.509	
	20		13.478	6.676	
	20.5		13.819	6.844	
	21		14.160	7.011	
	21.5		14.501	7.178	
	22		14.842	7.345	
	22.5		15.184	7.513	
	23		15.526	7.680	
	23.5		15.868	7.847	
	24		16.210	8.015	
	24.5		16.552	8.182	
	25		16.895	8.349	
	25.5		17.238	8.517	
	26		17.581	8.684	
	26.5		17.924	8.851	
	27		18.267	9.019	
	27.5		18.611	9.186	
	28		18.955	9.354	

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
	28.5		19.298	9.521	
	29		19.643	9.689	
	29.5		19.987	9.856	
	30		20.331	10.024	
10.5	10		6.701	3.335	5.532
	10.5		7.038	3.502	
	11		7.375	3.669	
	11.5		7.713	3.836	
	12		8.050	4.003	
	12.5		8.388	4.170	
	13		8.726	4.337	
	13.5		9.064	4.504	
	14		9.403	4.671	
	14.5		9.741	4.838	
	15		10.080	5.005	
	15.5		10.419	5.172	
	16		10.758	5.339	
	16.5		11.097	5.506	
	17		11.437	5.673	
	17.5		11.777	5.840	
	18		12.116	6.008	
	18.5		12.457	6.175	
	19		12.797	6.342	
	19.5		13.137	6.509	
	20		13.478	6.676	
	20.5		13.819	6.844	
	21		14.160	7.011	
	21.5		14.501	7.178	
	22		14.842	7.345	
	22.5		15.184	7.513	
	23		15.526	7.680	
	23.5		15.868	7.847	
	24		16.210	8.015	
	24.5		16.552	8.182	
	25		16.895	8.349	
	25.5		17.238	8.517	
	26		17.581	8.684	
	26.5		17.924	8.851	
	27		18.267	9.019	
	27.5		18.611	9.186	
	28		18.955	9.354	
	28.5		19.298	9.521	
	29		19.643	9.689	
	29.5		19.987	9.856	
	30		20.331	10.024	
11	10		6.701	3.335	5.841
	10.5		7.038	3.502	

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
	11		7.375	3.669	
	11.5		7.713	3.836	
	12		8.050	4.003	
	12.5		8.388	4.170	
	13		8.726	4.337	
	13.5		9.064	4.504	
	14		9.403	4.671	
	14.5		9.741	4.838	
	15		10.080	5.005	
	15.5		10.419	5.172	
	16		10.758	5.339	
	16.5		11.097	5.506	
	17		11.437	5.673	
	17.5		11.777	5.840	
	18		12.116	6.008	
	18.5		12.457	6.175	
	19		12.797	6.342	
	19.5		13.137	6.509	
	20		13.478	6.676	
	20.5		13.819	6.844	
	21		14.160	7.011	
	21.5		14.501	7.178	
	22		14.842	7.345	
	22.5		15.184	7.513	
	23		15.526	7.680	
	23.5		15.868	7.847	
	24		16.210	8.015	
	24.5		16.552	8.182	
	25		16.895	8.349	
	25.5		17.238	8.517	
	26		17.581	8.684	
	26.5		17.924	8.851	
	27		18.267	9.019	
	27.5		18.611	9.186	
	28		18.955	9.354	
	28.5		19.298	9.521	
	29		19.643	9.689	
	29.5		19.987	9.856	
	30		20.331	10.024	
11.5	10		6.701	3.335	6.150
	10.5		7.038	3.502	
	11		7.375	3.669	
	11.5		7.713	3.836	
	12		8.050	4.003	
	12.5		8.388	4.170	
	13		8.726	4.337	
	13.5		9.064	4.504	

360

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
	14		9.403	4.671	
	14.5		9.741	4.838	
	15		10.080	5.005	
	15.5		6.701	3.335	
	16		7.038	3.502	
	16.5		7.375	3.669	
	17		7.713	3.836	
	17.5		8.050	4.003	
	18		8.388	4.170	
	18.5		8.726	4.337	
	19		9.064	4.504	
	19.5		9.403	4.671	
	20		9.741	4.838	
	20.5		10.080	5.005	
	21		10.419	5.172	
	21.5		10.758	5.339	
	22		11.097	5.506	
	22.5		11.437	5.673	
	23		11.777	5.840	
	23.5		12.116	6.008	
	24		12.457	6.175	
	24.5		12.797	6.342	
	25		13.137	6.509	
	25.5		13.478	6.676	
	26		13.819	6.844	
	26.5		14.160	7.011	
	27		14.501	7.178	
	27.5		14.842	7.345	
	28		15.184	7.513	
	28.5		15.526	7.680	
	29		15.868	7.847	
	29.5		16.210	8.015	
	30		16.552	8.182	
12	10		6.701	3.335	6.459
	10.5		7.038	3.502	
	11		7.375	3.669	
	11.5		7.713	3.836	
	12		8.050	4.003	
	12.5		8.388	4.170	
	13		8.726	4.337	
	13.5		9.064	4.504	
	14		9.403	4.671	
	14.5		9.741	4.838	
	15		10.080	5.005	
	15.5		10.419	5.172	
	16		10.758	5.339	
	16.5		11.097	5.506	

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
	17		11.437	5.673	
	17.5		11.777	5.840	
	18		12.116	6.008	
	18.5		12.457	6.175	
	19		12.797	6.342	
	19.5		13.137	6.509	
	20		13.478	6.676	
	20.5		13.819	6.844	
	21		14.160	7.011	
	21.5		14.501	7.178	
	22		14.842	7.345	
	22.5		15.184	7.513	
	23		15.526	7.680	
	23.5		15.868	7.847	
	24		16.210	8.015	
	24.5		16.552	8.182	
	25		16.895	8.349	
	25.5		17.238	8.517	
	26		17.581	8.684	
	26.5		17.924	8.851	
	27		18.267	9.019	
	27.5		18.611	9.186	
	28		18.955	9.354	
	28.5		19.298	9.521	
	29		19.643	9.689	
	29.5		19.987	9.856	
	30		20.331	10.024	
12.5	10		6.701	3.335	6.768
	10.5		7.038	3.502	
	11		7.375	3.669	
	11.5		7.713	3.836	
	12		8.050	4.003	
	12.5		8.388	4.170	
	13		8.726	4.337	
	13.5		9.064	4.504	
	14		9.403	4.671	
	14.5		9.741	4.838	
	15		10.080	5.005	
	15.5		10.419	5.172	
	16		10.758	5.339	
	16.5		11.097	5.506	
	17		11.437	5.673	
	17.5		11.777	5.840	
	18		12.116	6.008	
	18.5		12.457	6.175	
	19		12.797	6.342	
	19.5		13.137	6.509	

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
	20		13.478	6.676	
	20.5		13.819	6.844	
	21		14.160	7.011	
	21.5		14.501	7.178	
	22		14.842	7.345	
	22.5		15.184	7.513	
	23		15.526	7.680	
	23.5		15.868	7.847	
	24		16.210	8.015	
	24.5		16.552	8.182	
	25		16.895	8.349	
	25.5		17.238	8.517	
	26		17.581	8.684	
	26.5		17.924	8.851	
	27		18.267	9.019	
	27.5		18.611	9.186	
	28		18.955	9.354	
	28.5		19.298	9.521	
	29		19.643	9.689	
	29.5		19.987	9.856	
	30		20.331	10.024	
13	10	0.783	7.311	3.638	5.705474
	10.5		7.679	3.821	
	11		8.047	4.003	
	11.5		8.415	4.185	
	12		8.783	4.367	
	12.5		9.151	4.549	
	13		9.520	4.731	
	13.5		9.889	4.914	
	14		10.258	5.096	
	14.5		10.628	5.278	
	15		10.997	5.460	
	15.5		11.367	5.643	
	16		11.737	5.825	
	16.5		12.107	6.007	
	17		12.478	6.189	
	17.5		12.848	6.372	
	18		13.219	6.554	
	18.5		13.590	6.737	
	19		13.961	6.919	
	19.5		14.333	7.101	
	20		14.704	7.284	
	20.5		15.076	7.466	
	21		15.448	7.649	
	21.5		15.820	7.831	
	22		16.193	8.014	
	22.5		16.566	8.196	

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
	23		16.938	8.379	
	23.5		17.312	8.561	
	24		17.685	8.744	
	24.5		18.058	8.926	
	25		18.432	9.109	
	25.5		18.806	9.291	
	26		19.180	9.474	
	26.5		19.555	9.657	
	27		19.929	9.839	
	27.5		20.304	10.022	
	28		20.679	10.205	
	28.5		21.054	10.388	
	29		21.430	10.570	
	29.5		21.805	10.753	
	30		22.181	10.936	
13.5	10		7.311	3.638	5.96507
	10.5		7.679	3.821	
	11		8.047	4.003	
	11.5		8.415	4.185	
	12		8.783	4.367	
	12.5		9.151	4.549	
	13		9.520	4.731	
	13.5		9.889	4.914	
	14		10.258	5.096	
	14.5		10.628	5.278	
	15		10.997	5.460	
	15.5		11.367	5.643	
	16		11.737	5.825	
	16.5		12.107	6.007	
	17		12.478	6.189	
	17.5		12.848	6.372	
	18		13.219	6.554	
	18.5		13.590	6.737	
	19		13.961	6.919	
	19.5		14.333	7.101	
	20		14.704	7.284	
	20.5		15.076	7.466	
	21		15.448	7.649	
	21.5		15.820	7.831	
	22		16.193	8.014	
	22.5		16.566	8.196	
	23		16.938	8.379	
	23.5		17.312	8.561	
	24		17.685	8.744	
	24.5		18.058	8.926	
	25		18.432	9.109	
	25.5		18.806	9.291	

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
	26		19.180	9.474	
	26.5		19.555	9.657	
	27		19.929	9.839	
	27.5		20.304	10.022	
	28		20.679	10.205	
	28.5		21.054	10.388	
	29		21.430	10.570	
	29.5		21.805	10.753	
	30		22.181	10.936	
14	10		7.311	3.638	6.224665
	10.5		7.679	3.821	
	11		8.047	4.003	
	11.5		8.415	4.185	
	12		8.783	4.367	
	12.5		9.151	4.549	
	13		9.520	4.731	
	13.5		9.889	4.914	
	14		10.258	5.096	
	14.5		10.628	5.278	
	15		10.997	5.460	
	15.5		11.367	5.643	
	16		11.737	5.825	
	16.5		12.107	6.007	
	17		12.478	6.189	
	17.5		12.848	6.372	
	18		13.219	6.554	
	18.5		13.590	6.737	
	19		13.961	6.919	
	19.5		14.333	7.101	
	20		14.704	7.284	
	20.5		15.076	7.466	
	21		15.448	7.649	
	21.5		15.820	7.831	
	22		16.193	8.014	
	22.5		16.566	8.196	
	23		16.938	8.379	
	23.5		17.312	8.561	
	24		17.685	8.744	
	24.5		18.058	8.926	
	25		18.432	9.109	
	25.5		18.806	9.291	
	26		19.180	9.474	
	26.5		19.555	9.657	
	27		19.929	9.839	
	27.5		20.304	10.022	
	28		20.679	10.205	
	28.5		21.054	10.388	

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
	29		21.430	10.570	
	29.5		21.805	10.753	
	30		22.181	10.936	
14.5	10		7.311	3.638	6.48426
	10.5		7.679	3.821	
	11		8.047	4.003	
	11.5		8.415	4.185	
	12		8.783	4.367	
	12.5		9.151	4.549	
	13		9.520	4.731	
	13.5		9.889	4.914	
	14		10.258	5.096	
	14.5		10.628	5.278	
	15		10.997	5.460	
	15.5		11.367	5.643	
	16		11.737	5.825	
	16.5		12.107	6.007	
	17		12.478	6.189	
	17.5		12.848	6.372	
	18		13.219	6.554	
	18.5		13.590	6.737	
	19		13.961	6.919	
	19.5		14.333	7.101	
	20		14.704	7.284	
	20.5		15.076	7.466	
	21		15.448	7.649	
	21.5		15.820	7.831	
	22		16.193	8.014	
	22.5		16.566	8.196	
	23		16.938	8.379	
	23.5		17.312	8.561	
	24		17.685	8.744	
	24.5		18.058	8.926	
	25		18.432	9.109	
	25.5		18.806	9.291	
	26		19.180	9.474	
	26.5		19.555	9.657	
	27		19.929	9.839	
	27.5		20.304	10.022	
	28		20.679	10.205	
	28.5		21.054	10.388	
	29		21.430	10.570	
	29.5		21.805	10.753	
	30		22.181	10.936	
15	10		7.311	3.638	6.743855
	10.5		7.679	3.821	
	11		8.047	4.003	

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
	11.5		8.415	4.185	
	12		8.783	4.367	
	12.5		9.151	4.549	
	13		9.520	4.731	
	13.5		9.889	4.914	
	14		10.258	5.096	
	14.5		10.628	5.278	
	15		10.997	5.460	
	15.5		11.367	5.643	
	16		11.737	5.825	
	16.5		12.107	6.007	
	17		12.478	6.189	
	17.5		12.848	6.372	
	18		13.219	6.554	
	18.5		13.590	6.737	
	19		13.961	6.919	
	19.5		14.333	7.101	
	20		14.704	7.284	
	20.5		15.076	7.466	
	21		15.448	7.649	
	21.5		15.820	7.831	
	22		16.193	8.014	
	22.5		16.566	8.196	
	23		16.938	8.379	
	23.5		17.312	8.561	
	24		17.685	8.744	
	24.5		18.058	8.926	
	25		18.432	9.109	
	25.5		18.806	9.291	
	26		19.180	9.474	
	26.5		19.555	9.657	
	27		19.929	9.839	
	27.5		20.304	10.022	
	28		20.679	10.205	
	28.5		21.054	10.388	
	29		21.430	10.570	
	29.5		21.805	10.753	
	30		22.181	10.936	

Scenario 2 - Design considering present AND future markets

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
15	10	0.783	7.311	3.638	6.744
	10.5		7.679	3.821	
	11		8.047	4.003	
	11.5		8.415	4.185	
	12		8.783	4.367	
	12.5		9.151	4.549	
	13		9.520	4.731	
	13.5		9.889	4.914	
	14		10.258	5.096	
	14.5		10.628	5.278	
	15		10.997	5.460	
	15.5		11.367	3.638	
	16		11.737	3.821	
	16.5		12.107	4.003	
	17		12.478	4.185	
	17.5		12.848	4.367	
	18		13.219	4.549	
	18.5		13.590	4.731	
	19		13.961	4.914	
	19.5		14.333	5.096	
	20		14.704	7.284	
	20.5		15.076	5.460	
	21		15.448	5.643	
	21.5		15.820	5.825	
	22		16.193	6.007	
	22.5		16.566	6.189	
	23		16.938	6.372	
	23.5		17.312	6.554	
	24		17.685	6.737	
	24.5		18.058	6.919	
	25		18.432	9.109	
	25.5		18.806	7.284	
	26		19.180	7.466	
	26.5		19.555	7.649	
	27		19.929	7.831	
	27.5		20.304	8.014	
	28		20.679	8.196	
	28.5		21.054	8.379	
	29		21.430	10.570	
	29.5		21.805	8.744	

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
	30		22.181	10.936	
20	10	0.898	8.385	4.173	6.697
	10.5		8.806	4.382	
	11		9.228	4.591	
	11.5		9.650	4.799	
	12		10.073	5.008	
	12.5		10.496	5.217	
	13		10.918	5.426	
	13.5		11.342	5.635	
	14		11.765	5.844	
	14.5		12.189	6.053	
	15		12.612	6.262	
	15.5		13.036	6.471	
	16		13.461	6.680	
	16.5		13.885	6.889	
	17		14.310	7.098	
	17.5		14.735	7.308	
	18		15.160	7.517	
	18.5		15.586	7.726	
	19		16.012	7.935	
	19.5		16.438	8.144	
	20		16.864	8.353	
	20.5		17.290	8.563	
	21		17.717	8.772	
	21.5		18.144	8.981	
	22		18.571	9.191	
	22.5		18.999	9.400	
	23		19.426	9.609	
	23.5		19.854	9.819	
	24		20.282	10.028	
	24.5		20.711	10.237	
	25		21.139	10.447	
	25.5		21.568	10.656	
	26		21.997	10.866	
	26.5		22.427	11.075	
	27		22.856	11.285	
	27.5		23.286	11.494	
	28		23.716	11.704	
	28.5		24.147	11.913	
	29		24.577	12.123	
	29.5		25.008	12.332	
	30		25.439	12.542	
25	10	0.998	9.318	4.638	6.659
	10.5		9.787	4.870	
	11		10.256	5.102	
	11.5		10.725	5.334	
	12		11.195	5.566	

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
	12.5		11.664	5.798	
	13		12.134	6.030	
	13.5		12.605	6.263	
	14		13.075	6.495	
	14.5		13.546	6.727	
	15		14.017	6.960	
	15.5		14.488	7.192	
	16		14.960	7.424	
	16.5		15.432	7.657	
	17		15.904	7.889	
	17.5		16.376	8.121	
	18		16.849	8.354	
	18.5		17.322	8.586	
	19		17.795	8.819	
	19.5		18.268	9.051	
	20		18.742	9.284	
	20.5		19.216	9.516	
	21		19.690	9.749	
	21.5		20.164	9.981	
	22		20.639	10.214	
	22.5		21.114	10.447	
	23		21.590	10.679	
	23.5		22.065	10.912	
	24		22.541	11.145	
	24.5		23.017	11.377	
	25		23.493	11.610	
	25.5		23.970	11.843	
	26		24.447	12.076	
	26.5		24.924	12.308	
	27		25.402	12.541	
	27.5		25.879	12.774	
	28		26.357	13.007	
	28.5		26.836	13.240	
	29		27.314	13.473	
	29.5		27.793	13.706	
	30		28.272	13.939	
30	10	1.087	7.313	3.639	14.523
	10.5		7.681	3.822	
	11		8.049	4.004	
	11.5		8.417	4.186	
	12		8.785	4.368	
	12.5		9.154	4.550	
	13		9.523	4.733	
	13.5		9.892	4.915	
	14		10.261	5.097	
	14.5		10.630	5.279	

V (m³)	P (MPa)	R (m)	Ts (mm)	Th (mm)	Lo (m)
	15		11.000	5.462	
	15.5		14.488	7.192	
	16		14.960	7.424	
	16.5		15.432	7.657	
	17		15.904	7.889	
	17.5		16.376	8.121	
	18		16.849	8.354	
	18.5		17.322	8.586	
	19		17.795	8.819	
	19.5		18.268	9.051	
	20		18.742	9.284	
	20.5		19.216	9.516	
	21		19.690	9.749	
	21.5		20.164	9.981	
	22		20.639	10.214	
	22.5		21.114	10.447	
	23		21.590	10.679	
	23.5		22.065	10.912	
	24		22.541	11.145	
	24.5		23.017	11.377	
	25		23.493	11.610	
	25.5		23.970	11.843	
	26		24.447	12.076	
	26.5		24.924	12.308	
	27		25.402	12.541	
	27.5		25.879	12.774	
	28		26.357	13.007	
	28.5		26.836	13.240	
	29		27.314	13.473	
	29.5		27.793	13.706	
	30		28.272	13.939	

371

APPENDIX C

JAVA CODE FOR THE HAND EXERCISER EXAMPLE PROBLEM

In this appendix, the computer program for the hand exerciser example is presented. The program is written in Java. A brief description of the classes and functions written is followed by the actual code. The latter part of the code is the program used in iSIGHT for results using optimization techniques.

The solution algorithm used for this example is as explained in Section 4.3.2. A brief description of the classes and functions are as follows:

Classes

handexerciser	This is the main class of the program for uncertain demand. From this class different functions are called. The solution algorithm is implemented in this class.
cDetailsOPv4	This class helps to write the output. i.e., detailed specifications of the product families into a file.

Functions

handexerciser	This constructor is comprised of the main solution algorithm
getwire	To get the cost, diameter and strength of different standard wires
demand	To define the forces demanded in the market space
initarray3	To initialize the flag
normCost	To calculate the normalized cost for the entire market space

normVar	To calculate the normalized variance for the entire market space
fCheck	To check if one number is a multiple of other
zc	To calculate the deviation function helping to make compromise
info1	To print the detailed specifications of platforms in the file
summary	To print the key factors of each platform such as cost, variance.
initCost	To initialize the cost
initVar	To initialize variables such as count

HandExerciser

```java
import java.io.*;
import java.math.*;

public class handexerciser {
    public static void main (String[] args)
    {
        new handexerciser();
    }

    double narray[][];
    double f=10;

    public handexerciser()
    {

        double w=90; //width
        double h=100;//height

        double E=200000.0;
        double rho=7860.0*1E-9;

        //To display output
        cResult4 Results3 = new cResult4();
        cDetailsOPv4 Details = new cDetailsOPv4();

// 1 Force
// 2 d
// 3 Flag
// 4 k
// 5 D
// 6 Demand
// 7 mt
// 8 N
// 9 H
```

```
// 10L

          int dflag=0; //to check if all springs satisfy constraints
          int bestcount=0;

          narray = new double[101][12]; //storage of all springs
          double wire[][];
          wire = new double[16][5]; //storage of all wire information

          double ztemp=100;

//Demand
     for (int i=0; i<101; i++)
     {
          narray[i][6]=5000.0;
     }


          demand();

getwire();

//Initialize variables
double maxf=100.0;


double CmatTotal=0.0;
double Cmat =0.0;
double dF2=0.0;
double F2=0;
double F1=0;
double dF1=0.0;
double Fmax;
double R=100.0;
double Cpack=0.0;

int count=0;
int countTotal=0;
//initCost();
//initvar();

double pd=0;
double dem[];

//array used to calculate cost of mandrels
dem = new double[101];
for (int i=0; i<101; i++)
{
     dem[i]=0.0;
}

double ctool[];
ctool = new double[101];
int md=0;
double ctd=0.0;
double cto=0.0;
double cmand=500.0;//cost of mandrel
```

```
int nw=1;
double Cinv=0;
double d=0.0;
double cw=0.0;
double mt=0.0;
double N=0.0;
double H=0.0;
double L=0.0;
double sy=0.0;
double su=0.0;
double se=0.0;


double Ctemp=100000000.0;
double Cgrand=0.0;
double fdF1=0.0;
double fdF2=0.0;
double CtoolTotal=0.0;

double vd[];
vd = new double[5];
double vsqg[];
vsqg = new double[5];
double vsgrand=0.0;
double vcmat[];
vcmat = new double[5];
double vCgrand[];
vCgrand = new double[5];
double varCgrand=0.0;
double vde=0.0;
double cd=0.0;
double varn=0.0;
double Cgrandn=0.0;
double z=0.0;

int tp2=0;


for(int n=0; n<101; n++)
{
      narray[n][3]=0.0;
}

      for(int j=1; j<15; j++)
      {
      for(int p=1; p<15; p++)
      {
      for(int q=1; q<15; q++)
      {
            for(int r=1; r<15; r++)
            {

            //cost of inventory depending on number of wires
            if ((j==p)&&(j==q)&&(j==r)) Cinv=0;
            if ((j!=p)&&(j!=q)&&(j!=r)) Cinv=6000;
            if ((j==p)&&(j==q)&&(j!=r)) Cinv=2000;
            if ((j==p)&&(j!=q)&&(j==r)) Cinv=2000;
```

```
if ((j!=p)&&(j==q)&&(j==r)) Cinv=2000;
if ((j==p)&&(j!=q)&&(j!=r)&&(q==r)) Cinv=2000;
if ((j==p)&&(j!=q)&&(j!=r)&&(q!=r)) Cinv=4000;

if ((j!=p)&&(j==q)&&(j!=r)&&(p==r)) Cinv=2000;
if ((j!=p)&&(j==q)&&(j!=r)&&(p!=r)) Cinv=4000;

if ((j!=p)&&(j!=q)&&(j==r)&&(p==q)) Cinv=2000;
if ((j!=p)&&(j!=q)&&(j==r)&&(p!=q)) Cinv=4000;

for (int l=1; l<51; l++)
{
initarray3();
demand();    //Force initialization
F2 = narray[l][1];
dF2 = F2 - narray[0][1];
int dFnode = l; //number of nodes in dF1

//Check if dF2 is a multiple of maxf
int fc = fCheck(maxf, dF2);

if (fc > 0)
{;}
else
{
//number of elements
int Felement = (100)/dFnode;
//loop through elements and assign each node
for (int g=0; g<101; g++)
{
ctool[g]=0.0;
}

int di=0;

for (int m=1; m<Felement+1; m++)
{
di=0;
for (int g=0; g<101; g++)
{
dem[g]=0.0;
}
Fmax = narray[m*dFnode][1];
//fill previous nodes
for (int n=1; n<((m*dFnode)+1); n++)
{
if(narray[n][3]<1.0)
{
narray[n][1] = Fmax;
narray[n][3] = 1.0;
dem[di] = dem[di]+narray[n][6];
}
else {;}
}

//calculatation of cost of mandrels
md = (int)(dem[di]/58400)+1;
```

```
pd = dem[di]/md;
ctd = cmand*pd*md/50000;
//System.out.println(pd*md/50000);

cto = 0.1*cmand*md*(1-(pd/58400));
ctool[di] = ctool[di]  + ctd + cto;
di++;


}

CtoolTotal = ctool[di-1];
count=0;
dflag=0;

for(int s=0; s<101; s++)
{
d=wire[j][1];
narray[s][2] = d;
sy=wire[j][2];
//assignment of wires to nodes
if((s>25)&&(s<=50))
{
d=wire[p][1];
narray[s][2] = d;
sy=wire[p][2];
}
if((s>50)&&(s<=75))
{
d=wire[q][1];
narray[s][2] = d;
sy=wire[q][2];
}
if(s>75)
{
d=wire[r][1];
narray[s][2] = d;
sy=wire[r][2];
}
R=100.0;
narray[s][4]=(sy*3.142*d*d*d)/(32*narray[s][1]*R); //K
if ((narray[s][4]>1.7)&&(narray[s][4]<2)) sy=sy-600;
if (narray[s][4]>2) sy=sy-1000;
narray[s][4]=(sy*3.142*d*d*d)/(32*narray[s][1]*R);
if ((narray[s][4]<1.00001)&&(narray[s][4]>0.7))
     R=R-35;
if ((narray[s][4]<0.7))
     R=R-60;
narray[s][4]=(sy*3.142*d*d*d)/(32*narray[s][1]*R);
if ((narray[s][4]>1.0)&&(narray[s][4]<1.6)) count++;
     double a=4*narray[s][4]-4;
double b=-4*d*narray[s][4]-d;
double c=d*d;

narray[s][5]=(-b+Math.sqrt(b*b-4*a*c))/(2*a); //D
//variance given to D
vd[2]=narray[s][5]*0.8;
vd[1]=narray[s][5]*0.9;
```

```
            vd[0]=narray[s][5];
            vd[3]=narray[s][5]*1.1;
            vd[4]=narray[s][5]*1.2;

            }
            countTotal = countTotal + count;
            if (count==100) dflag=1;
            for (int b=1; b<51; b++)
            {
                    initarray3();
                    demand();
                    F1 = narray[b][1];
                    dF1 = F1 - narray[0][1];
                    int dFnode1 = b; //number of nodes in dF1
                    //check if dF1 is a multiple of dF2
                    int fc1 = fCheck(dF2, dF1);
                    if (fc1 > 0)
                    {;}
                    else
                    {

                    int Felement1 = (100)/dFnode1; //number of nodes

                    for (int b1=1; b1<Felement1+1; b1++)
                    {
                    Fmax = narray[b1*dFnode1][1];
                    //fill previous nodes with Fmax if they haven't been
    filled
                    for (int n=0; n<((b1*dFnode1)+1); n++)
                    {
                    if(narray[n][3]<1.0)
                    {
                    narray[n][1] = Fmax;
                    narray[n][3] = 1.0;
                    }
                    else {;}
                    }
                    }

                    CmatTotal=0;
                    for(int g=0; g<5; g++)
                    {
                    vcmat[g]=0.0;
                    }
                    for(int m=0; m<5; m++)  //number of levels for
    variance
                    {
                    for(int t=0; t<101; t++) //loop for all springs
                    {
                    H=h+vd[m];
                    //System.out.println("H"+H+"D"+narray[t][5]);
                    mt=Math.atan(2*H/narray[t][5]);
                    double d4 = ((int)d)*((int)d)*((int)d)*((int)d);

                    N=d4*E*(3.142-2*mt)/(68*vd[m]*100*narray[t][1]);
                    double tp = (68*vd[m]*vd[m]*100*N/(d4*E));
                    H=(vd[m]/2)*Math.tan(0.5*(3.142-tp));
```

```
                //System.out.println("H"+H);
                L=1.2*(3.142*vd[m]*N+2*Math.sqrt(((w-vd[m])/2)*((w-
vd[m])/2)+H*H));
                cw = wire[j][3]; //cost of wire
                Cmat = narray[t][6]*cw*rho*3.142*d*d*L/4; //cost of
material of a spring
                CmatTotal = CmatTotal + Cmat; //total cost of
material
                narray[t][7] = mt;
                narray[t][8] = N;
                narray[t][9] = H;
                narray[t][10] = L;
                }// t loop
                vcmat[m]=CmatTotal;

                }//m loop
                //cost of packing
                Cpack=350*Felement1;
                //total cost
                Cgrand=vcmat[0] + CtoolTotal + Cinv+Cpack;

                //calculation of variance
                vsgrand=0.0;
                for(int g=0; g<5; g++)
                {
                vCgrand[g]=0.0;
                }

                vCgrand[0]=Cgrand;
                for(int g=1; g<5; g++)
                {
                vCgrand[g]=vcmat[g] + CtoolTotal + Cinv;
                vsgrand=vsgrand+((vCgrand[g]-vCgrand[0])*(vCgrand[g]-
vCgrand[0]))/3;
                }

                //normalize
                //varn = normCost(Cgrand, Cgrandn);
                //Cgrandn = normVar(vsgrand, varn);
                varn=5E8/vsgrand;

                Cgrandn=29000/Cgrand;
                vde=1-varn;
                cd=1-Cgrandn;

                //zc = z(0.5, 0.5, cd, vde)
                z=1*vde+0*cd;
                if (dflag==1)
                {
                //Details.info1(j, p, q, r, dF2, dF1, narray);
                tp2=tp2+1;
                Details.summary(j, p, q, r, dF2, dF1, vcmat[0],
CtoolTotal, Cgrand, Cinv, Cpack, vde, z);
                }

                //System.out.println(Cgrand);
```

```
                              if ((z<ztemp)&&(dflag==1))
//&&(dF1!=1))&&(CmatTotal>20000))
                        {
                        //Ctemp=Cgrand;
                        ztemp=z;
                        fdF1=dF1;
                        fdF2=dF2;
                        Results3.confile(fdF2, fdF1, vcmat[0], CtoolTotal,
Cgrand, varn, Cgrandn, z);
                        Results3.confile(j, p, fdF2, fdF1, vcmat[0],
CtoolTotal, Cgrand, vsgrand, varn, Cgrandn, z);
                        Results3.confile(dF2, dF1, 0, 0,0,0,0,0,0);
                        Details.info1(j, p, q, r, dF2, dF1, narray);
                        Details.summary(j, p, q, r, dF2, dF1, vcmat[0],
CtoolTotal, Cgrand, Cinv, Cpack, vde, z);
                        bestcount++;
                        }
                        }//else

                        }//b loop
                        }//else
                        }//l loop

              }//r
              }//q
              }//p
              }//j

       }//end of exerciser1


       public int fCheck(double n, double m)
       {
              Double x = new Double(n/m);
              int y = x.intValue();
              double z = y+0.0;
              double fractioncheck = (n/m) - z;
              int flag = 0;
              if (fractioncheck < 0 || fractioncheck > 0)
              {
                     flag = 1;
              }
              return flag;
       }//end of fCheck()

       public void initarray3()
       {
              for (int i=0; i<101; i++)
              {
                     narray[i][3]=0.0;
              }
       }


       public double normCost(double Cgrand, double Cgrandn)
       {
              Cgrandn=29000/Cgrand;
```

```java
                return Cgrandn;
        }


        public double normVar(double vsgrand, double varn)
        {
                varn=5E8/vsgrand;
                return varn;
        }


        public double z(double w1, double w2, double cd, double vde)
        {
                double zc=0.0;
                zc = w1*cd + w2*vde;
                return zc;
        }



        //Force initialization
        public void demand()
        {
                f=10;
                for (int i=0; i<101; i++)
                {
                narray[i][1]=f;
                f=f+1;
                }
        }


        public void initcost()
        {
                CmatTotal=0.0;
                Cmat =0.0;
                Cpack = 0.0;
        }


        public void initvar()
        {
                Count=0;
                countTotal=0;
                F1=0;
                F2=0;
        }

public double[][] getwire()
{
wire[1][1]=0.79; wire[1][2]=1978;
wire[2][1]=1.30; wire[2][2]=1816;
wire[3][1]=1.60; wire[3][2]=1756;
wire[4][1]=1.91; wire[4][2]=1702;
wire[5][1]=2.59; wire[5][2]=1618;
wire[6][1]=3.18; wire[6][2]=1565;
wire[7][1]=4.11; wire[7][2]=1493;
wire[8][1]=0.89; wire[8][2]=1678;
wire[9][1]=1.22; wire[9][2]=1559;
wire[10][1]=1.60; wire[10][2]=1499;
wire[11][1]=2.69; wire[11][2]=1349;
wire[12][1]=3.18; wire[12][2]=1319;
```

```
wire[13][1]=4.17; wire[13][2]=1139;
wire[14][1]=4.88; wire[14][2]=1330;
wire[15][1]=5.26; wire[15][2]=1310;

wire[1][3]=3.53;
wire[2][3]=2.48;
wire[3][3]=2.39;
wire[4][3]=2.37;
wire[5][3]=2.22;
wire[6][3]=2.18;
wire[7][3]=2.16;
wire[8][3]=2.42;
wire[9][3]=2.26;
wire[10][3]=2.15;
wire[11][3]=1.36;
wire[12][3]=0.91;
wire[13][3]=0.89;
wire[14][3]=1.25;
wire[15][3]=1.15;
}

}
```

cDetailsOPv4

```
 import java.io.*;

 public class cDetailsOPv4
 {
      BufferedWriter out;
      public cDetailsOPv4()
      {
           try
           {
                out = new BufferedWriter(new
FileWriter("Details.txt"));
                out.write("\n");
           }
           catch (IOException e)
           {
                System.err.println("FileStream error " + e);
           }
      }//end of cDetailsOPv4

//puts detailed info in file
public void info1(int j, int p, int q, int r, double fdF2, double fdF1,
double[][] narray)
      {
           try
           {

                out.write("\n");

                out.write("j: " + j);

                out.write("   ");
                out.write("p: " + p);
```

```
                    out.write("    ");
                    out.write("q: " + q);
                    out.write("    ");
                    out.write("r: " + r);
                    out.write("    ");
                    out.write("fdF2: " + fdF2);
                    //out.write(fdF2);
                    out.write("    ");
                    out.write("fdF1:   " + fdF1);
                    //out.write(fdF1);
                    out.write("\n");

out.write("Node       Force                    d                     Flag
k                         D                     Demand
mt                 N                            H
L");
                    out.write("\n");

                    for (int i=0; i<101; i++)
                    {
                            Integer h = new Integer(i);
                            String sh = h.toString();
                            out.write(sh);
                            out.write("         ");

                            for (int g= 1; g<11; g++)
                            {
                                    Double d = new Double(narray[i][g]);
                                    String s = d.toString();
                                    out.write(s);
                                    out.write("     ");
                            }
                            out.write("\n");
                    }
            }
            catch (IOException e)
            {
                    System.err.println("FileStream error " + e);
            }
    }//end info1()

    //summary of a run
    public void summary(int j, int p, int q, int r, double dF2,
double dF1, double CmatTotal, double CtoolTotal, double Cgrand, double
Cinv, double Cpack, double vsgrand, double z)
    {
            try
            {
                    out.write("j = " + j + " p = " + p + " q = " + q + "
r = " + r + " dF2 = " + dF2 + " dF1 = " + dF1);
                    out.write("\n");

                    out.write("CmatTotal = $" + CmatTotal);
                    out.write("\n");
```

383

```
                        out.write("CtoolTotal = $" + CtoolTotal);
                        out.write("\n");

                        out.write("Cpack = $" + Cpack);
                        out.write("\n");

                        out.write("Cinv = $" + Cinv);
                        out.write("\n");

                        out.write("Cgrand = $" + Cgrand);
                        out.write("\n");

                        out.write("vsgrand = " + vsgrand);
                        out.write("\n");

                        out.write("z = " + z);
                        out.write("\n");

                        out.write("\n");

                }
                catch (IOException e)
                {
                        System.err.println("FileStream error " + e);
                }
        }

        //close file
        public void close()
        {
                try
                {
                        out.close();
                }
                catch (IOException e)
                {
                        System.err.println("FileStream error " + e);
                }
        }//end close()

 }//end of cDetailsOPv4
```

iSIGHT

```
import java.io.*;
import java.math.*;

public class handexisight {
      public static void main (String[] args)
      {
              new handexisight();
      }

      double narray[][];
      double f=10;

      public handexisight()
```

```
{
try
{

        double idf1;
        String idf1s;
        double idf2;
        String idf2s;

        double ij;
        String ijs;
        double ip;
        String ips;
        double iq;
        String iqs;
        double ir;
        String irs;

            //read input file
        FileReader ss = new FileReader("subspace.txt");
        BufferedReader in = new BufferedReader (ss);
        idf2s = in.readLine();
        idf2 = Double.valueOf(idf2s).doubleValue();
        System.out.println(idf2);

        idf1s = in.readLine();
        idf1 = Double.valueOf(idf1s).doubleValue();
        System.out.println(idf1);

        ijs = in.readLine();
        ij = Double.valueOf(ijs).doubleValue();
        System.out.println(ij);

        ips = in.readLine();
        ip = Double.valueOf(ips).doubleValue();
        System.out.println(ip);

        iqs = in.readLine();
        iq = Double.valueOf(iqs).doubleValue();
        System.out.println(iq);

        irs = in.readLine();
        ir = Double.valueOf(irs).doubleValue();
        System.out.println(ir);


        double w=90;
        double h=100;

        double E=200000.0;
        double rho=7860.0*1E-9;

        double F=55.0;
        cResult3 Results3 = new cResult3();
        cDetailsOP Details = new cDetailsOP();
```

```
// 1 Force
// 2 d
// 3 Flag
// 4 k
// 5 D
// 6 Demand
// 7 mt
// 8 N
// 9 H
// 10L

            int dflag=0;
            int bestcount=0;

            narray = new double[101][12];
            double wire[][];
            wire = new double[16][5];
            double ztemp=100;

//Demand
      for (int i=0; i<101; i++)
      {
            narray[i][6]=5000.0;
      }

            initarray1();

wire[1][1]=0.79; wire[1][2]=1978;
wire[2][1]=1.30; wire[2][2]=1816;
wire[3][1]=1.60; wire[3][2]=1756;
wire[4][1]=1.91; wire[4][2]=1702;
wire[5][1]=2.59; wire[5][2]=1618;
wire[6][1]=3.18; wire[6][2]=1565;
wire[7][1]=4.11; wire[7][2]=1493;
wire[8][1]=0.89; wire[8][2]=1678;
wire[9][1]=1.22; wire[9][2]=1559;
wire[10][1]=1.60; wire[10][2]=1499;
wire[11][1]=2.69; wire[11][2]=1349;
wire[12][1]=3.18; wire[12][2]=1319;
wire[13][1]=4.17; wire[13][2]=1139;
wire[14][1]=4.88; wire[14][2]=1330;
wire[15][1]=5.26; wire[15][2]=1310;

wire[1][3]=3.53;
wire[2][3]=2.48;
wire[3][3]=2.39;
wire[4][3]=2.37;
wire[5][3]=2.22;
wire[6][3]=2.18;
wire[7][3]=2.16;
wire[8][3]=2.42;
wire[9][3]=2.26;
wire[10][3]=2.15;
wire[11][3]=1.36;
wire[12][3]=0.91;
wire[13][3]=0.89;
wire[14][3]=1.25;
```

```
wire[15][3]=1.15;

double maxf=100.0;
double CmatTotal=0.0;
double Cmat =0.0;
double dF2=0.0;
double F2=0;
double F1=0;
double dF1=0.0;
double Fmax;
double R=100.0;
double Cpack=0.0;

int count=0;
int countTotal=0;


double pd=0;
double dem[];
dem = new double[101];

for (int i=0; i<101; i++)
{
      dem[i]=0.0;
}

double ctool[];
ctool = new double[101];
int md=0;
double ctd=0.0;
double cto=0.0;
double cmand=500.0;
int nw=1;
double Cinv=0;
double d=0.0;
double cw=0.0;
double mt=0.0;
double N=0.0;
double H=0.0;
double L=0.0;
double sy=0.0;
double su=0.0;
double se=0.0;


double Ctemp=10000000.0;
double Cgrand=0.0;
double fdF1=0.0;
double fdF2=0.0;
double CtoolTotal=0.0;

double vd[];
vd = new double[5];
double vsqg[];
vsqg = new double[5];
double vsgrand=0.0;
double vcmat[];
```

387

```
vcmat = new double[5];
double vCgrand[];
vCgrand = new double[5];
double varCgrand=0.0;
double vde=0.0;
double cd=0.0;
double varn=0.0;
double Cgrandn=0.0;
double z=0.0;

        for(int n=0; n<101; n++)
        {
        narray[n][3]=0.0;
        }

        int j= (int) ij;
        int p= (int) ip;
        int q= (int) iq;
        int r= (int) ir;

        d=wire[j][1];

        initarray3();
        initarray1();     //Force initialization
        dF2=idf2;
        int dFnode = (int) idf2; //number of nodes in dF1

        int fc = factorCheck(maxf, dF2);
        if (fc > 0)
        {;}
        else
        {

        //number of elements
        int Felement = (100)/dFnode;
        //loop through elements and assign each node
        for (int g=0; g<101; g++)
        {
        ctool[g]=0.0;
        }

        int di=0;

        for (int m=1; m<Felement+1; m++)
        {
                di=0;
                for (int g=0; g<101; g++)
                {
                dem[g]=0.0;
                }
                Fmax = narray[m*dFnode][1];
                //fill previous nodes
                for (int n=1; n<((m*dFnode)+1); n++)
                {
                if(narray[n][3]<1.0)
                {
                narray[n][1] = Fmax;
```

388

```
narray[n][3] = 1.0;
dem[di] = dem[di]+narray[n][6];
}
else {;}
}
md = (int)(dem[di]/58400)+1;

pd = dem[di]/md;
ctd = cmand*pd*md/50000;
cto = 0.1*cmand*md*(1-(pd/58400));
ctool[di] = ctool[di]  + ctd + cto;
di++;


}
CtoolTotal = ctool[di-1];

count=0;
dflag=0;

for(int s=0; s<101; s++)
{
d=wire[j][1];
narray[s][2] = d;
sy=wire[j][2];
if((s>25)&&(s<=50))
{
d=wire[p][1];
narray[s][2] = d;
sy=wire[p][2];
}

if((s>50)&&(s<=75))
{
d=wire[q][1];
narray[s][2] = d;
sy=wire[q][2];
}
if(s>75)
{
d=wire[r][1];
narray[s][2] = d;
sy=wire[r][2];
}
R=100.0;
narray[s][4]=(sy*3.142*d*d*d)/(32*narray[s][1]*R); //K
if ((narray[s][4]>1.7)&&(narray[s][4]<2)) sy=sy-600;
if (narray[s][4]>2) sy=sy-1000;
narray[s][4]=(sy*3.142*d*d*d)/(32*narray[s][1]*R);
if ((narray[s][4]<1.00001)&&(narray[s][4]>0.7))
       R=R-35;
if ((narray[s][4]<0.7))
       R=R-60;
narray[s][4]=(sy*3.142*d*d*d)/(32*narray[s][1]*R);
if ((narray[s][4]>1.0)&&(narray[s][4]<1.6)) count++;

double a=4*narray[s][4]-4;
double b=-4*d*narray[s][4]-d;
```

```
double c=d*d;
narray[s][5]=(-b+Math.sqrt(b*b-4*a*c))/(2*a); //D

vd[2]=narray[s][5]*0.8;
vd[1]=narray[s][5]*0.9;
vd[0]=narray[s][5]*1.2;
vd[3]=narray[s][5]*1.1;
vd[4]=narray[s][5]*1.2;

}
countTotal = countTotal + count;
if (count==100) dflag=1;

dF1=idf1;

initarray3();
initarray1();
int dFnode1 = (int) idf1; //number of nodes in dF1

int fc1 = factorCheck(dF2, dF1);
if (fc1 > 0)
{;}
else
{

int Felement1 = (100)/dFnode1;

for (int b1=1; b1<Felement1+1; b1++)
{
Fmax = narray[b1*dFnode1][1];
//fill previous nodes with Vmax if they haven't been filled
yet
for (int n=0; n<((b1*dFnode1)+1); n++)
{
if(narray[n][3]<1.0)
{
narray[n][1] = Fmax;
narray[n][3] = 1.0;
}
else {;}

}
}
CmatTotal=0;
for(int g=0; g<5; g++)
{
vcmat[g]=0.0;
}
for(int m=0; m<5; m++)
{
for(int t=0; t<101; t++)
{
H=h+vd[m];
//System.out.println("H"+H+"D"+narray[t][5]);
mt=Math.atan(2*H/narray[t][5]);
                                    double d4 =
((int)d)*((int)d)*((int)d)*((int)d);
```

390

```
                                              if (d4==0) d4=1;
           N=d4*E*(3.142-2*mt)/(68*vd[m]*100*narray[t][1]);
           double tp = (68*vd[m]*vd[m]*100*N/(d4*E));

        H=(vd[m]/2)*Math.tan(0.5*(3.142-tp));
           //System.out.println("H"+H);
           L=1.2*(3.142*vd[m]*N+2*Math.sqrt(((w-vd[m])/2)*((w-
vd[m])/2)+H*H));
           cw = wire[j][3];
           Cmat = narray[t][6]*cw*rho*3.142*d*d*L/4;
           CmatTotal = CmatTotal + Cmat;

           narray[t][7] = mt;
           narray[t][8] = N;
           narray[t][9] = H;
           narray[t][10] = L;
           }// t loop
           vcmat[m]=CmatTotal;
        }//m loop

           Cpack=350*Felement1;
           Cgrand=vcmat[0] + CtoolTotal + Cinv+Cpack;

           vsgrand=0.0;
           for(int g=0; g<5; g++)
           {
           vCgrand[g]=0.0;

           }
           vCgrand[0]=Cgrand;
           for(int g=1; g<5; g++)
           {
           vCgrand[g]=vcmat[g] + CtoolTotal + Cinv;
           vsgrand=vsgrand+((vCgrand[g]-vCgrand[0])*(vCgrand[g]-
vCgrand[0]))/3;
           }

           varn=5E8/vsgrand;
           Cgrandn=29000/Cgrand;
           vde=1-varn;
           cd=1-Cgrandn;
           z=0.5*vde+0.5*cd;
           ztemp=z;
           fdF1=dF1;
           fdF2=dF2;

           bestcount++;
        }//else
        }//else
        }
        catch (IOException e)
        {
           System.err.println("FileStream error " + e);
        }

        }//end of handexisight
}
```

APPENDIX D

# OPTIMIZATION RESULTS OF THE HAND EXERCISER EXAMPLE

The results of the exhaustive search method for the hand exerciser example are compared with some optimization techniques. The combination of optimization techniques used as discussed in Section 4.2.4 are as follows:

- Genetic Algorithms and Simulated Annealing

First Case:

	Start	End
Genetic Algorithms	$\Delta f3=100(15\ 15\ 15\ 15)$	$\Delta f3=50(11\ 11\ 13\ 13)$
	$\Delta f2=10$	$\Delta f2=50$
	$\Delta f1=5$	$\Delta f1=2$
	$Z=1$	$Z=0.796466$
Simulated Annealing	$\Delta f3=50(11\ 11\ 13\ 13)$	$\Delta f3=50(11\ 11\ 6\ 6)$
	$\Delta f2=50$	$\Delta f2=50$
	$\Delta f1=2$	$\Delta f1=25$
	$Z=0.796466$	$Z=0.583595$
	$\Delta f3=50(11\ 11\ 6\ 6)$	$\Delta f3=50(15\ 15\ 6\ 6)$
	$\Delta f2=50$	$\Delta f2=50$
	$\Delta f1=25$	$\Delta f1=25$
	$Z=0.583595$	$Z=0.512011$

Genetic Algorithms-Simulated Annealing Results 1
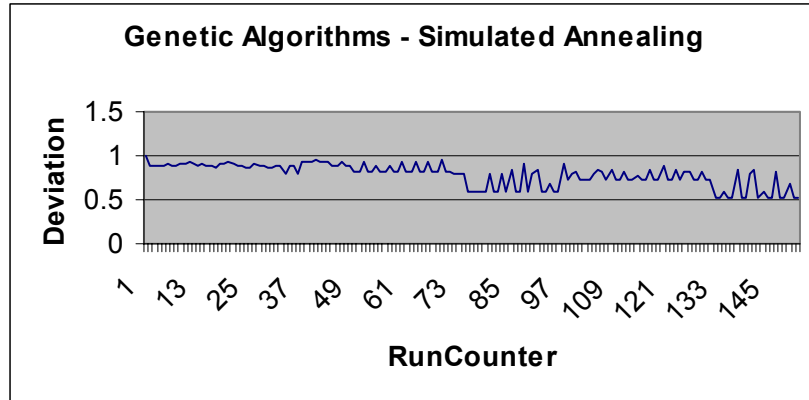
Deviation vs Run Counter Results 1

Second Case (Different Starting Point):

	Start	End
Genetic Algorithms	$\Delta f3=100(1\ 1\ 1\ 1)$	$\Delta f3=50(4\ 4\ 2\ 2)$
	$\Delta f2=25$	$\Delta f2=25$
	$\Delta f1=1$	$\Delta f1=25$
	$Z=1$	$Z=0.730344$
Simulated Annealing	$\Delta f3=50(4\ 4\ 2\ 2)$	$\Delta f3=50(7\ 7\ 2\ 2)$
	$\Delta f2=25$	$\Delta f2=50$
	$\Delta f1=25$	$\Delta f1=10$
	$Z=0.730344$	$Z=0.659194$

Genetic Algorithms-Simulated Annealing Results 2
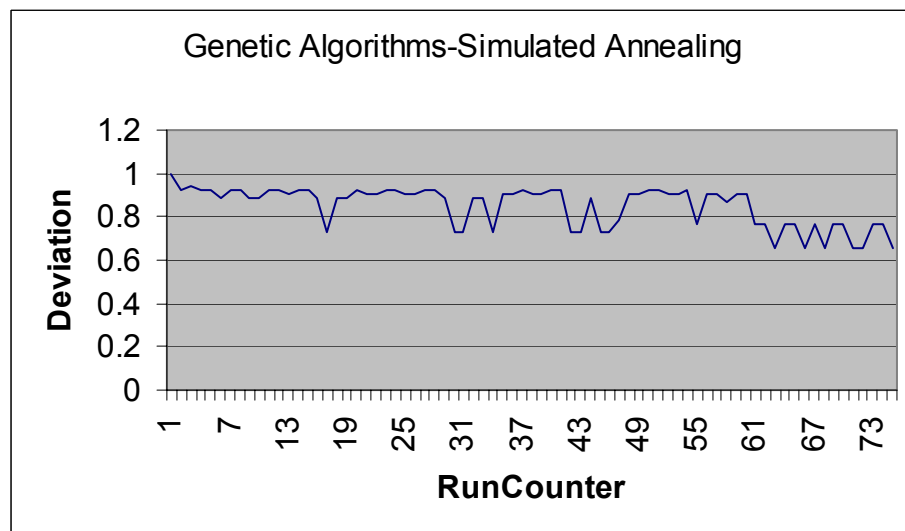
Deviation vs Run Counter Results 2

- Mixed Integer Programming and Genetic Algorithms

	Start	End
Mixed Integer Programming-Genetic Algorithms	$\Delta f3=100(15\ 15\ 15\ 15)$	$\Delta f3=50(15\ 15\ 13\ 13)$
	$\Delta f2=50$	$\Delta f2=50$
	$\Delta f1=25$	$\Delta f1=25$
	Z=0.891936	Z=0.717896
	$\Delta f3=50(15\ 15\ 13\ 13)$	$\Delta f3=50(15\ 15\ 6\ 6)$
	$\Delta f2=50$	$\Delta f2=50$
	$\Delta f1=25$	$\Delta f1=25$
	Z=0.717896	Z=0.51201

Mixed Integer Programming-Genetic Algorithms Results
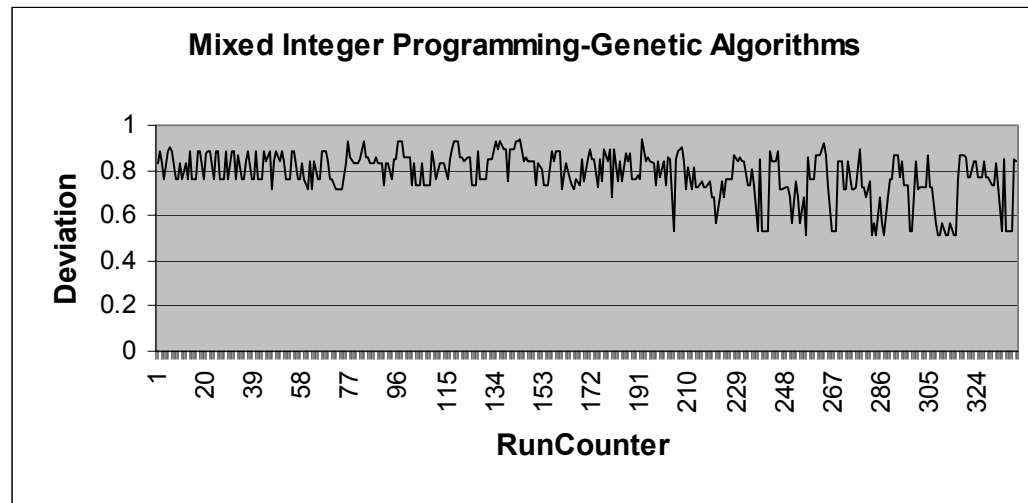
Deviation vs Run Counter

- *Sequential Quadratic Programming*

Max. Number of Iterations	40
Relative Gradient Step	0.0001
Abs. Obj. Convergence	0.0001

	Start	End
SQP	$\Delta f3=50(15\ 15\ 6\ 6)$	$\Delta f3=50(15\ 15\ 6\ 6)$
	$\Delta f2=50$	$\Delta f2=50$
	$\Delta f1=2$	$\Delta f1=10$
	$Z=0.584478$	$Z=0.503384$

Sequential Quadratic Programming Results
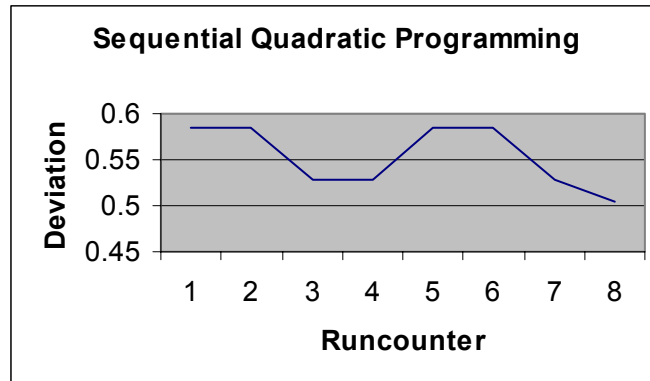
Deviation vs Run Counter

REFERENCES

Abramson, D., Abela, J., 1992, "A parallel genetic algorithm for solving the school timetabling problem," *Proceedings of the Fifteenth Australian Computer Science Conference (ACSC-15),* Vol. 14, pp. 1-11.

Anderson, D. M., 1997, *Agile Product Develpment for Mass Customization*, Irwin Professional Publishing, Chicago.

Bednar, H. H., 1986, *Pressure Vessel Design Handbook*, Van Nostrand Reinhold, NY.

Bejan, A., 1996, "Street Network Theory of Organization in Nature," *Journal of Advanced Transportation,* Vol. 255, No. 7, pp. 85 - 107.

Bejan, A., 1997, *Advanced Engineering Thermodynamics*, John Wiley & Sons, New York.

Bejan, A., 2000, *Shape and Structure: From Engineering to Nature*, Cambridge University Press.

Belegundu, A. D. and Chandrupatla, T. R., 1999, *Optimization Concepts and Applications in Engineering*, Prentic-Hall, Inc., Upper Saddle River, NJ.

Byrne, D. M. and Taguchi, S., 1987, "The Taguchi Approach to Parameter Design," *40th Annual Quality Congress Transactions*, pp. 168-177.

Carone, M. J., Williams, C. B., Allen, J. K. and Mistree, F., 2003, "An Application of Constructal Theory in the Multi-Objective Design of Product Platforms," *Proceedings of the 2003 ASME Design Engineering Technical Conferences*, Chicago, Ill. Paper Number DETC2003/DTM-48667.

Chamberlain, M. K., Williams, C. B., Mistree, F., 2004, "Strategic Design: Leveraging Market Forecasts and Emerging Technologies in Engineering Design Applications," *The 10th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Albany, NY. Paper Number AIAA-2004-4652.

Chang, T. S., Ward, A. C., Lee, J. and Jacox, E. H., 1994, "Conceptual Robustness in Simultaneous Engineering: An Extension of Taguchi's Parameter Design," *Research in Engineering Design,* Vol. 6, pp. 211-222.

Chen, W., Allen, J. K., Tsui, K.-L. and Mistree, F., 1996, "A Procedure for Robust Design: Minimizing Variations Caused by Noise Factors and Control Factors," *ASME Jounal of Mechanical Design,* Vol. 118, No. 4, pp. 478-485.

Chen, W., Mavris, D. N. and Mistree, F., 1996, "A Concept Exploration Method for Determining Robust Top-Level Specifications," *Engineering Optimization,* Vol. 26, No. 2, pp. 137-158.

Chen, W., Sahai, A., Messac, A., Sundararaj, G.J., 2000, "Exploration of the Effectiveness of Physical Programming in Robust Design," *ASME Journal of Mechanical Design,* Vol. 122, No. 2, pp. 155-163.

Chen, W., Wiecek, M. and Zhang, J., 1999, "Quality Utility: A Compromise Programming Approach to Robust Design," *ASME Journal of Mechanical Design,* Vol. 121, No. 2, pp. 179-187.

Collier, D. A., 1981, "The Measurement and Operating Benefits of Component Part Commonality," *Decision Sciences,* Vol. 12, No. 1, pp. 85-96.

Conway, C. L. and Tiry, J. M., 1998, "Mass Customization Comes to Hoffman," *Agility & Global Competition,* Vol. 2, No. 2, pp. 16-23.

Courtois, P. J., 1985, "On Time and Space Decomposition of Complex Structures," *Communications of the ACM,* Vol. 28, No. 6, pp. 580-603.

Dahmus, J. B., J. P. Gonzalez-Zugasti, K. N. Otto, 2000, "Modular Product Architecture," *ASME Design Theory and Methodology Conference*, Baltimore, MD. Paper Number DETC2000/DTM-14565.

Engineous, July 2003, "i-SIGHT Software", http://www.engineous.com.

Ericsson, A. and Erixon, G., 1999, *Controlling Design Variants: Modular Platforms*, ASME Press, Dearborn, MI.

Fernández, M. G., 2002, *On Decision Support for Distributed Collaborative Design and Manufacturing*, M.S. Thesis, G.W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA.

Fernández, M. G., Seepersad, C. C., Rosen, D. W., Allen, J. K. and Mistree, F., 2001, "Utility-Based Decision, Support for Selection in Engineering Design," *13th International Conference on Design Theory and Methodology*, Pittsburgh, PA. Paper Number DETC2001/DAC-21106.

Goldberg, D. E., 1989, *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison-Wesley, Reading, MA.

Goldberg, D. E., Deb K., 1991, "A comparative analysis of selection schemes used in genetic algorithms," *Foundations of Genetic Algorithms,* Vol. 1, pp. 69-93.

Gonzalez-Zugasti, J. P., Otto, K. N., 2000, "Modular Platform-Based Product Family Design," *ASME Advances in Design Automation Conference*, Baltimore, MD. Paper Number DETC2000/DAC-14238.

Hayter, A. J., 2002, *Probability and Statistics for Engineers and Scientists*, Thomson Learning, Pacific Grove, CA.

Hazelrigg, G. A., 1996, *Systems Engineering and Approach to Information-Based Design*, Prentice Hall.

Hernandez, G., 2001, *Platform Design for Customizable Products as a Problem of Access in Geometric Space*, Ph.D. Dissertation, G.W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA.

Hernandez, G., Allen, J. K. and Mistree, F., 2002, "Design of Hierarchic Platforms for Customizable Products," *ASME Design Automation Conference*, Montreal. Paper Number DETC2002/DAC-34095.

Hernandez, G., Allen, J. K. and Mistree, F., 2003, "Design of Platforms for Customizable Products as a Problem of Access in a Geometric Space," *Journal of Engineering Optimization,* Vol. 35, No. 3, pp. 229-254.

Hernandez, G., Allen, J.K. and Mistree, F., 2003, "A Theory and Method for Combining Multiple Approaches for Product Customization," *Second Interdisciplinary Congress on Mass Customization and Personalization*, Technische Universität München, Munich, Germany.

Iyer, H. V., Krishnamurty, S., 1998, "A Preference-Based Robust Design Metric," *ASME Design Technical Conference*, Atlanta, GA. Paper Number DAC5625.

Jacobs, O. L. R., 1967, *An Introduction to Dynamic Programming: The Theory of Multistage Decision Processes*, Chapman and Hall, London.

Keeney, R. L. and Raiffa, H., 1976, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, John Wiley & Sons, New York.

Kirkpatrick, S., Gelatt, C. D. Jr, "Optimization by simulated annealing," in *Readings in computer vision: issues, problems, principles, and paradigms*, pp. 606 - 615, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., (1987).

Koch, P. N., 1997, *Hierarchical Modeling and Robust Synthesis for the Preliminary Design of Large-Scale Complex Systems*, PhD Dissertation, G.W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA.

Kota, S., and Sethuraman K., 1998, "Managing Variety in Product Families through Design for Commonality," *ASME Design Theory and Methodology Conference*, Atlanta, GA. Paper Number DETC98/DTM-5651.

Lehnerd, A. P., 1987, "Revitalizing the Manufacture and Design of Mature Global Product," *Technology and Global Industry: Companies and Nations in the World Economy*, pp. 49-64.

McDermott, C. M., Stock G. N., 1994, "The Use of Common Parts and Designs in High-Tech Industries: A Strategic Approach," *Production and Inventory Management Journal,* Vol. 35, No. 3, pp. 17-28.

Messac, A., 1996, "Physical Programming: Effective Optimization for Computational Design," *AIAA Journal,* Vol. 34, No.1, pp. 149-158.

Meyer, M. H., 1997, "Revitalize Your Product Lines through Continuous Platform Renewal," *Research Technology Management,* Vol. 40, No. 2, pp. 17-28.

Meyer, M. H., Lehnerd, A. P., 1997, "The Power of Product Platforms: Building Value and Cost Leadership," Free Press, NY.

Mistree, F., Hughes, O. F. and Bras, B. A., "The Compromise Decision Support Problem and the Adaptive Linear Programming Algorithm," in *Structural Optimization: Status and Promise*, pp. 247-286, (*M. P. Kamat, eds.*), Washington, D.C.: AIAA, (1993).

Mistree, F., Seepersad, C. C., Dempsey, B. M., McDowell, D. L. and Allen, J. K., 2002, "Robust Concept Exploration Methods in Materials Design," *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, GA. Paper Number AIAA-2002-5568.

Mistree, F., Smith, W. F. and Bras, B. A., "A Decision-Based Approach to Concurrent Engineering," in *Handbook of Concurrent Engineering*, pp. 127-158, (*H. R. Paresai and W. Sullivan, eds.*), New York: Chapman & Hall, (1993).

Myers, R. H. and Montgomery, D. C., 1995, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, John Wiley and Sons, New York.

Nair, V. N., 1992, "Taguchi's Parameter Design: A Panel Discussion," *Technometrics,* Vol. 34, pp. 127-161.

Nayak, R. U., Chen, W., Allen, J. K. and Mistree, F., 1999, "A Variation-Based Methodology for Product Family Design," *Proceedings of the 2000 ASME Design Automation Conference*, Baltimore, MD. Paper Number DETC2000/DAC-8876.

Nelson, S. A., Parkinson, M. B. and Papalambros, P. Y., 2001, "Multicriteria Optimization in Product Platform Design," *Journal of Mechanical Design,* Vol. 123, No. 2, pp. 199-204.

Panchal, J. H., Chamberlain, M. K., Rosen, D. W., Allen, J. K. and Mistree, F., 2002, "A Service Based Architecture for Information and Asset Utilization in Distributed Product Realization," *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, GA. Paper Number AIAA-2002-5480.

Pederson, K., Emblemsvag, J., Allen, J. K. and Mistree, F., 2000, "Validating Design Methods and Research - The Validation Square," *ASME Design Theory and Methodology Conference*, Baltimore, MA. Paper Number DETC00/DTM-14579.

Phadke, M. S., 1989, *Quality Engineering Using Robust Design*, Prentice Hall, Englewood Cliffs, NJ.

Pine, B. J., 1993, *Mass Customization: The New Frontier in Business Competition*, Harvard Business School Press, Boston.

Ramakrishnan, B., Rao, S.S., 1991, "A Robust Optimization Approach Using Taguchi's Loss Function for Solving Nonlinear Optimization Problems," *Advances in Design Automation, ASME,* DE-32-1, pp. 241-248.

Rosen, D. W., 1996, "Design of Modular Product Architectures in Discrete Design Spaces Subject to Life Cycle Issues," *ASME Advances in Design Automation Conference*, Irvine, CA. Paper Number DETC96/DAC-1485.

Rothwell, R., Gardiner P., "Robustness and Product Design Families," in *Design Management: A Handbook of Issues and Methods (M. Oakley, Ed.)*, pp. 279-292, Basil Blackwell Inc., Cambridge, MASons, New York, (1990).

Seepersad, C. C., 2001, *The Utility-Based Compromise Decision Support Problem with Applications in Product Platform Design*, The G.W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA.

Seepersad, C. C., Cowan F.S., Chamberlain M.K., Mistree F., 2002, "Strategic Design: Leveraging and Innovation for a Changing Marketplace," *Computer-BasedDesign: Engineering Design Conference 2002*, King's College London, UK.

Seepersad, C. C., Hernandez, G. and Allen, J. K., 2000, "A Quantitative Approach to Determining Product Platform Extent," *ASME Advances in Design Automation Conference*, Baltimore, MD. Paper Number DETC2000/DAC-14288.

Seepersad, C. C., Mistree, F. and Allen, J. K., 2002, "A Quantitative Approach for Designing Multiple Product Platforms for an Evolving Portfolio of Products," *ASME Design Engineering Technical Conferences, Advances in Design Automation*, Montreal, Canada. Paper Number DETC2002/DAC-34096.

Siddique, Z., Rosen D. W., 2001, "On Combinatorial Design Spaces for the Configuration Design of Product Families," *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing,* Vol. 15, No. 2, pp. 91-108.

Siddique, Z. and Rosen, D. W., 2000, "Product Family Configuration Reasoning Using Discrete Design Spaces," *ASME Design Theory and Methodology Conference*, Baltimore, MD. Paper Number DETC2000/DTM-14666.

Simon, H. A., 1996, *Sciences of the Artificial*, The MIT Press, Cambridge, MA.

Simon, H. A. and Ando, A., 1961, "Aggregation of Variables in Dynamic Systems," *Econometrica*, No. 29, pp. 111-138.

Simpson, T. W., 1999, *A Concept Exploration Method for Product Family Design*, Ph. D. Dissertation, G. W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta.

Simpson, T. W., 2003, "Product Platform Design and Optimization: Status and Promise," *ASME Design Engineering Technical Conferences*, Chicago, Ill. Paper Number DETC2003/DAC-48717.

Simpson, T. W., Chen, W., Allen, J. K. and Mistree, F., 1996, "Conceptual Design of a Family of Products Through the Use of the Robust Concept Exploration Method," *6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Bellevue, WA. Paper Number AIAA96-4161.

Simpson, T. W., Chen, W., Allen, J. K. and Mistree, F., "Use of the Robust Concept Exploration Method to Facilitate the Design of a Family of Products," in *Simultaneous Engineering: Methodologies and Applications*, pp. 247-278, (*U. Roy and J. M. Usher, eds.*), New York: Chapman-Hall, (1999).

Simpson, T. W., Maier, J.R.A. and F. Mistree, 1999, "A Product Platform Concept Exploration Method for Product Family Design," *ASME Design Theory and Methodology Conference*, Las Vegas, NV. Paper Number DETC99/DTM-8761.

Steuer, R. E., 1986, *Multiple Criteria Optimization: Theory, Computation and Application*, John Wiley, New York.

Suh, N. P., 1990, *Principles of Design*, Oxford University Press, Oxford, U.K.

Sundaresan, S., Ishii, K. and Houser, D.R., 1993, "A Robust Optimization Procedure with Variations on Design Variables and Constraints,," *Advances in Design Automation, ASME,* DE-Vol. 69-1, pp. 379-386.

Tseng, M. and Jiao, J., 1998, "Design for Mass Customization by Developing Product Family Architecture," *ASME Design for Manufacture Conference*, Atlanta, GA. Paper Number DETC98/DFM-5717.

Tsui, K.-L., 1992, "An Overview of Taguchi Method and Newly Developed Statistical Methods for Robust Design," *IIE Transactions,* Vol. 24, No. 5, pp. 44-57.

von Neumann, J., and Morgenstern., O., 1947, *Theory of Games and Economic Behavior*, Princeton University Press, Princeton, N.J.

Welch, W. J., Yu, T.-K., Kang, S. M. and Sacks, J., "Computer Experiments for Quality Control by Parameter Design," *Journal of Quality Technology,* Vol. 22, No. 1, pp. 15-22.

Williams, C., 2003, *Platform Design for Customizable Products and Processes with Non-Uniform Demand*, M.S. Thesis, The G.W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA.

Williams, C. W., Allen, J.K., Rosen, D.W., Mistree, F., 2004, "Designing Platforms for Customizable Products in Markets with Non-Uniform Demand," *Design Theory and Methodology Conference*, Salt Lake City, Utah. Paper Number ASME DETC2004/DTM-57469.

Yu, P. L., 1973, "A Class of Solutions for Group Decision Problems," *Management Science,* Vol. 19, pp. 936-946.

Zeleny, M., "Compromise programming," in *Multiple Criteria Decision Making*, pp. 262-301, (*J. L. Cochrane, Zeleny, M., eds.*), Columbia, SC: University of South Carolina Press, (1973).