PLATFORM BASED APPROACH FOR ECONOMIC PRODUCTION OF A PRODUCT FAMILY

by

ANAND M. CHOUBEY

B. S., National Institute of Foundry & Forge Technology, India, 2005

A THESIS

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Industrial & Manufacturing Systems Engineering
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2007

Approved by:

Major Professor
Dr. David Ben-Arieh

Abstract

In present competitive market, there is growing concern for ascertaining and fulfilling the individual customer's wants and needs. Therefore, the focus of manufacturing has been shifting from mass production to mass customization, which requires the manufacturers to introduce an increasing number of products with shorter life span and at a lower cost. Also, another challenge is to manage the variety of products in an environment where demands are stochastic and the lead times to fulfill those demands are short.

The focus of this thesis is to develop and investigate platform based production strategies, as opposed to producing each product independently, which would ensure the economic production of the broader specialized products with small final assembly time and under demand uncertainty.

The thesis proposes three different platform based production models. The first model considers the economic production of products based on a single platform and with forecasted demands of the products. The model is formulated as a general optimization problem that considers the minimization of total production costs.

The second model is the extension of the first model and considers the production of products based on multiple platforms and considers the minimization of total production costs and the setup costs of having multiple platforms.

The third model is also an extension of the first model and considers the demands of the products to be stochastic in nature. The model considers the minimization of total production costs and shortage costs of lost demands and holding cost of surplus platforms under demand uncertainties. The problem is modeled as a two stage stochastic programming with recourse.

As only the small instances of the models could be solved exactly in a reasonable time, various heuristics are developed by combining the genetic evolutionary search approaches and some operations research techniques to solve the realistic size problems. The various production models are validated and the performances of the various heuristics tailored for the applications are investigated by applying these solution approaches on a case of cordless drills.

Table of Contents

List of Figures

List of Tables

CHAPTER 1 - Introduction

This chapter provides the foundations for the principle objective of this dissertation, which is to develop platform-based approaches to facilitate the economic production of a given set of products (product family). The subject matter of this thesis falls in the intersection of several areas of current research interest. These include: (1) using platform strategies to produce products to realize mass customization, (2) formulating production planning problems as a general optimization problems and solving them using optimization techniques, and in particular, developing genetic evolutionary heuristics to efficiently solve the problems that are not solvable exactly in a reasonable amount of time, and (3) capturing uncertainty in the model.

Section 1.1 contains the motivation for the research. Section 1.2 provides the foundation and framework for investigating the proposed research, and the objective of this section is to establish context for the reader. In Section 1.3, the objectives and contribution for the work are described. Sections 1.1 -1.3 set the foundation for the chapters that follow, leading to the development of three production models: (1) economic production of a product family using single platform, (2) economic production of a product family using multiple platforms, and (3) economic production for a product family under demand uncertainty using single platform. Finally, Section 1.4 contains an overview of the dissertation.

1.1 Motivation

In today's highly volatile market there is growing concern for ascertaining and fulfilling the individual customer wants and needs. "The customers now have plenty of choice … they have become more aware…they select the product that most closely fulfills their opinion of being the best value for the money…" (Hollins and Pugh, 1990). Therefore, "customers can no longer be lumped together in a huge homogeneous market…" (Pine, 1993) rather this competitive world of manufacturing requires the manufacturer to introduce an increasing number of products with shorter life span and at a lower cost. This requires the producer to continuously seek ways to reduce the production costs, while still offering attractive products. In the past, a company could capture the market and enjoy high profits by producing large volume of the same model, as the case of the Ford Model T automobile. Now, the focus in manufacturing has been shifting from mass production to mass customization; a trend no longer limited to high value products. This phenomenon is demonstrated by the fact that from 1973 to 1989, there has been a 70% increase in the number of car models produced in the US with commensurate drop in the volume of production per model (McDuffie *et al.*, 1996).

In this marketing environment, in a company, the marketing management demands for the production of broader specialized product lines that would lead to higher market share. Whereas, the operations management predicts that the cost and complexity would increase when there is more product variety. In addition to that, one another challenge is to manage the variety of products in an environment where demands are stochastic and the lead times to fulfill those demands are short.

Consequently, companies are looking for strategies that would ensure the economic production of broader specialized products with small final assembly time and under demand uncertainty. Hence, the focus of this thesis is to develop and investigate such strategies.

Toward this end, the various strategies that have received significant attention in literature and practice includes, but not limited to the use of concepts from delayed differentiation (Lee 1996, Lee and Tang 1998, Swaminathan and Tayur 1998), exploiting commonality at the product design state (Ulrich and Pearson 1993, Hayes *et al.* 1998), use of lean manufacturing concepts (Womack *et al.* 1991), and product platform strategy (Meyer and Lehnerd 1997). The platform-based production strategy is very widely implemented strategy to create product families that provide sufficient variety for the market while maintaining economies of scale (Simpson 2003).

The platform approach offers the advantage of developing several variants from a unified platform with a considerable cost savings. Contrary to developing the product singly, when the products are developed as a part of a family, i.e. developing several variants from a platform, can result in considerable cost saving. Products based on platform architecture can be varied more easily by the introduction of new variant products without requiring the redesign of the whole product. Variant products make use of the product platform as the starting point and add or remove components to increase the number of features, performance, or variety of the base product. Also, in an environment where demands are stochastic, this approach facilitates storing of inventory in the form of semi-finished products on the basis of which the final products would be produced with small final assembly time. Hence, this approach can provide cheap inventory management and better response to customers while minimizing shortage and holding costs.

Because of its advantages, this approach has gained acceptance by many corporations as the means to increase their product count without a cost-per-part increase. Black and Decker's applied this idea to its power tool products (Meyer and Lehnerd, 1997). Volkswagen used platform architecture strategy and reduced development and production costs (Wilhelm, 1997). Sony applied this approach for its product development process (Sanderson and Uzumeri, 1995). AeroAstro Inc., used platform architecture with their multipurpose radio platform, and solved many of the communication problems faced by spacecraft system designers (Caffrey *et al*., 2002). HP's Ink jet printer platform architecture is rejuvenated constantly and hence the derivative products are constantly upgraded (Meyer, 1997).

In this thesis, we propose platform-based approaches for economic production of a product family for various problem models. The foundations for developing those approaches are presented in next section.

1.2 Research Focus – Research Issues and Hypotheses

This section provides the foundation and framework for investigating the proposed research, and the objective of this section is to establish context for the reader.

The objective of this dissertation is to provide production approaches based on using product platforms to ensure economic production of products. The research focus in this dissertation can be captured by presenting the issues that need to be addressed in the form of questions and providing answers in the form of hypotheses.

Research issue: *How mass customization can be realized using platform strategy?*

Hypothesis: The platforms can be mass produced and the products in a family can be produced by adding and/or removing some components from the platforms.

Research issue: *How customer responsiveness can be increased using platform strategy?*

Hypothesis: The products, when their demands are realized, can be produced by adding and/ or removing components to the platforms and hence the final assembly time is less.

Research issue: *How uncertainty in demand of the products is addressed?*

Hypothesis: The uncertainty in the model can be captured using stochastic programming.

Research issue: *How the overall platform-based production approach is realized?*
Hypothesis: The platform-based production approach can be modeled mathematically as a general optimization problem and can be solved using exact and/or heuristic approaches.

These research issues and the hypotheses are explored in detail in the upcoming chapters of this thesis. The resulting research contribution is presented in the next section.

1.3 Research Contributions

As mentioned earlier the subject matter of this thesis falls in the intersection of several areas of current interest. The contributions from the dissertation in the mentioned research areas are as follows.

The thesis proposes platform-based production approach for the economic production of a given product family. Three different platform-based production models are developed, which are as follows.

A single platform-based approach for the production of product family when the demands of the products are forecasted demand values. The model considers the minimization of total production cost that includes the costs of components, costs of mass assembly, and costs of adding/removing components from the individual products, while considering the individual demand and structure of each product type.

A multiple platform-based approach for the production of product family when the demands of the products are forecasted demand values. The model considers the production of products based on multiple platforms as opposed to single platform proposed in the first model and considers the minimization of total production costs considered in first model plus the setup costs of having platforms.

The third model is the single platform-based production approach for the production of product family when the demands of the products are stochastic demand values. The model consider the minimization of total production costs and shortage costs of lost demands and holding cost of leftover platforms under demand uncertainties. The problem is modeled as a two stage stochastic programming with recourse. The demand uncertainty is presented by two ways (1) considering various demand scenarios with associated probabilities and (2) considering probability distribution of demand of each product. By the use of small hypothetical cases various investigations on the properties of stochastic program are done.

All the three types of production planning models are mathematically formulated and formulations are improved by making non-linear constraints linear and by adding some valid constraints to make it solvable exactly in less time. However, the realistic size models could not be solved exactly in a reasonable amount of time. Therefore, various heuristics are developed by combining the genetic search approach and some operations research techniques to solve the real and large size instances of the problems. The heuristics tailored for the applications are validated by comparing the results obtained by the exact method and that of the heuristic approaches, and by applying these solution approaches on a case of cordless drills. The performances of various heuristic approaches are investigated; some sensitivity analyses on various parameters of the models and heuristics are done, and insights into the various proposed models are presented.

1.4 Overview of the Thesis

The objective of the introduction section was to provide the motivations, foundations and context which provides the basis for the specific contributions that are made in this research. The rest of the thesis is organized as follows.

Chapter 2 provides the background and related literature review on research areas such as platform based production, and evolutionary genetic search to solve large scale optimization problems. Various areas related to platform based production and design are reviewed, such as platform and product family based production approach, various streams of research in platform approaches, various platform strategies and platform techniques, and various optimization techniques used in solving platform-based production and design problems for various objectives.

Chapter 3 provides the overall problem description and assumptions, and the notations and nomenclatures used throughout the thesis.

Chapter 4 considers the problem of determining a platform for the production of a product family while minimizing the overall production cost. The problem is formulated as a general optimization problem of minimizing the production cost using platform architecture while satisfying the part assembly constraints. Both an optimal formulation and an evolutionary strategy based on Genetic Algorithm are presented. The approaches are illustrated with an example of a family of cordless drills presented in Section 3.3.

Chapter 5 considers the problem of proposing multiple platforms for the production of a given product family while minimizing the overall production cost. The methodology considers the demand for each product variant, with the decision variables as the optimal number of platforms, optimal configuration of each platform, and assignment of the products to the platforms. The problem is formulated as an integer program, and both the optimal formulation and an evolutionary strategy based on Genetic Algorithm are presented. The approach is illustrated with the example from a family of Cordless Drills.

Chapter 6 presents the third model where the demands are stochastic and the product family is produced based on single platform. The problem is formulated as a two stage stochastic programming model with recourse. The objective is to minimize the total production cost that includes the cost of production of platforms, cost of production of products using the platforms, holding cost for unused platforms and stock-out cost for lost demands. This problem is solved using three approaches, exact method, a genetic algorithm based heuristic that combines integer programming to solve the problem, and a pure probability based genetic search approach. The three approaches are investigated and their importance for various problem instances is presented. The approaches are illustrated and validated by using the same example of a family of cordless drills.

Chapter 7 is the final chapter and contains a summary of the thesis and recommendations for future work.

CHAPTER 2 - Background

2.1 Concept of Product Platform and Related Research

As concluded in the last chapter, a platform based production approach is used to increase variety, increase customer responsiveness by shortening the final assembly time, reduce overall production cost and hence, mass customization is realized. The following reviews the concepts of product platforms, related research, and various relevant issues.

2.1.1 Product Family and Product Platform

A product family is a set of products considered together for production as they share some common basic sets of attributes. A product family can be viewed as a set of variables such as components, functions or features, etc. that remain constant from product to product, and others that vary across the product line. The modification of features across the product line in a given family can be done by *scaling*, or by addition and/ or removal of modules and/ or components (Messac 2002a).

The product family considered here has stable core functionality, but has variability in secondary functions, which are successful in their market niches. Usually the product family has a long life cycle, which must adapt to a rapidly changing environment. The product family is produced using the concepts of platforms, utilizing the commonality between the products.

Ulrich and Eppinger (2003) define ***a platform*** as a collection of assets, including component designs, shared by multiple products. It can also be defined as a set of shared functionality, components, subsystems and manufacturing processes across the product family (Robertson and Ulrich 1998). More specifically, in this thesis, a platform is also considered to be a set of shared components among multiple products. A product from a product family is produced using a particular platform by adding or removing some of the components that are assembled using the particular platform.

The platform is mass produced and the product family is derived using the platforms. Figure 2.1 gives an example of product family and a platform.

Figure 2.1 A product platform and a product family

In this example, the product family constitutes drills, saws and a light. The product family shares some common components. The product platform, as shown in figure, could be a set of components such as, a chuck, a spindle, a battery, power switch, etc.

2.1.2 Various streams of research

Various streams of research in the area of product platforms is greatly influenced by contributions from Pine (1993) in the area of mass customization, Meyer and Lehnerd (1997) in the area of platform concepts, Sanderson and Uzmeri (1996) in the area of managing product families (Allada et al., 2006). Krishnan and Ulrich (2001), Simpson (2004), Jose and Tollenaere (2005), and more recently Allada *et al.* (2006) have provided a review of various aspects of product platform development methodologies. Simpson *et al.* (2006) provides an overview of the platform concept, application areas, and ongoing research and expanding views on platforms in academia and in industry.

There have been two streams of research in the area of product platform formation:

- Qualitative and/or conceptual model based approaches towards Platform development (management domain), and
- Quantitative model based approaches (engineering domain)

Qualitative approaches model the issues related to market share of the products derived based on platforms, desired financial performances, product introduction, etc. This type of approaches deals with the platform and product family planning problems at a high level of abstraction and with management perspective. For example, Maier and Fadel (2001) suggest a selection platform based on the appropriate product family design, which is based on product variability, and various market attributes such as market size, market type and number of target market niches. Dahmus *et al.* (2001) suggests an appropriate platform based on strategies such as price movements, cannibalization effects, optimal sales demand, etc. Shil and Allada (2005) provides a methodology for evaluation of risk neutral product portfolio. They evaluate product development projects by considering cannibalization effects and select the project with highest utility value. Wilson and Norton (1989) provide a framework to determine the optimal entry timing for a product line extension to protect against product cannibalization. Martin and Ishii (1996), Martin and Ishii (1997), and Martin and Ishii (2002) develop product platform architecture to gain competitive advantage by reducing the redesign efforts and time-to-market. Kota *et al.* (2000) present an objective measure to capture the level of component commonality in a product family. The underlying idea they propose is to minimize non-value added variations across models within a product family without limiting customer choices. Park and Simpson (2005) propose a cost estimation framework to support product family design.

Quantitative approaches focus on the engineering design and production aspect of products based on platforms. The research in this area can be further divided in two three categories:

- Scalable based platform formation problems
- Module based or configuration based platform formation problems
- Combination of both module based and scalable platform formation problems

Scale based product family design is a method by which some of the variables in a product family are kept fixed while other variables, scaling variables, are “stretched” or “shrink” to generate the variants within the product family. There are many examples in industry that have used platform scaling to develop the product family to satisfy various market niches. Black & Decker (Lehnerd 1987), Rolls Royce (Rothwell and Gardiner 1990), Boeing (sabbagh 1996), Honda (Naughton *et al.* 1997) have successfully applied this strategy to produce product family on the basis of platforms to satisfy target market niches. This area has been explored by various researchers. For example, Simpson *et al.* (2001) develop of a family of Universal electric motors by finding the values of common variable (platform variables) and the values of scaling variables, while minimizing the performance loss with respect to the individually optimized family of products. Messac *et al.* (2002b) determine the common variables and the scalable variables and their values and then decide the product variants around the platform. Hernandez *et al.* (2003) develop a scalable product platform with the objective to minimize the material cost, welding and forging cost, given the target specifications of the customizable product variants.

Module based product family design is a method by which the product family member are derived by adding and/ or removing modules from the platform. This approach is more prominent approach as this approach allows the platform leveraging for products from different market segments too. This approach used the concept of modularity in product design (Baldwin and Clark 2000, Ulrich and Eppinger 2000, Ericsson and Erixon 1999). There are many examples in industry that have used module or configuration based platform approaches to develop the product family to satisfy various market niches. Sony (Sanderson and Uzumeri 1997), Volkswagen (Wilhelm 1997), Nippondenso Co. Ltd. (Whitney 1995), Hewlett Packard (Feitzinger and Lee 1997) have successfully applied this strategy to produce a product family on the basis of platforms to satisfy target market niches. There has been plethora of research in this area. For instance, Fujita *et al.* (1999) develop a modular platform for a family of products while minimizing production cost, facility cost and material cost. Moor *et al.* (1999) proposes conjoint analysis to design modular product platform. Siddique and Rosen (2000) design platforms by exploiting assembly commonality of assembly processes of an existing set of products. Gonzales-Zugasti and Otto (2000) present an optimization approach for designing product families built on modular platforms. The method allows for the design of the modules that are shared across multiple members of the family, or becomes the part of the platform, as well as the variational modules. Many optimization approaches in platform based product development considers module or configuration based structure of product family and platforms. The research in this thesis falls in this category.

Some consider the combination of both module-based and scale-based platform formation strategy together. Fujita and Yoshida, 2001 proposes a simultaneous optimization method for both module combination and module attributes (design variables) of multiple products to form a platform. Three types of modules are identified; the modules whose design variables are common and have the same value are identified as common modules the modules whose design variables are similar in nature are identified as scalable modules, the third type of modules that are identified are the modules that lead to variety in the product family members. This methodology combines both module and scalable platform formation approaches.

Figure 2.2 presents the various streams of research in the area of platform formation approaches.

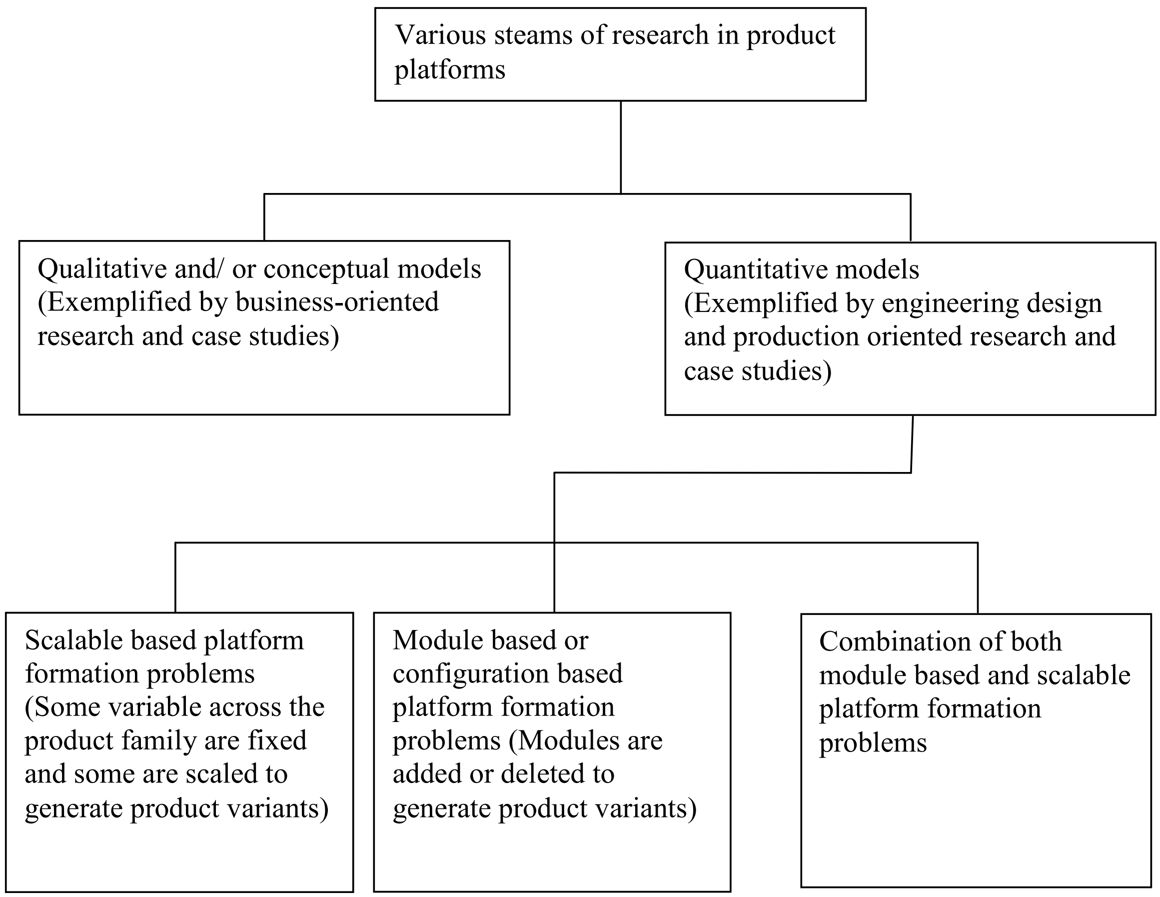


Figure 2.2 Various streams of research related to product platform formation problems

2.1.3 Platform Strategies

There are various strategies to implement platforms for the development or production of the products in the family. The various platform strategies that are used are Horizontal Leveraging, Vertical Leveraging, and Beachhead approach (Meyer and Lehnerd 1997). In vertical leveraging a platform is shared among the low-end, mid-range and high-end variants of a family. In horizontal leveraging a product family consists of the variants of a product in the same market segment. And, in case of the beachhead approach, the platform is shared between both the types of variants of the product. Figure 2.3 presents the various platform leveraging strategies.

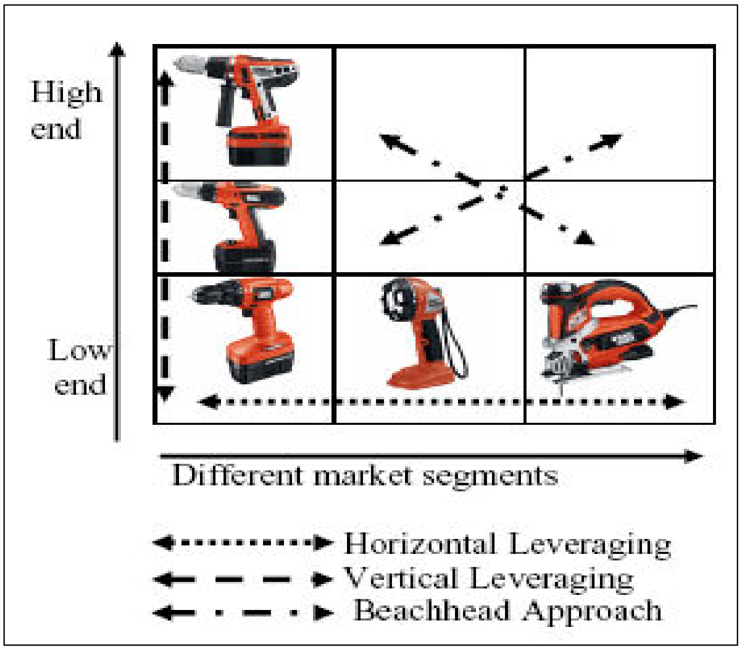


Figure 2.3 Platform leveraging strategies

Most horizontal leveraging approaches take advantage of modular platform, and most vertical leveraging take advantage of scalable platform (Simpson *et al.* 2001, Simpson 2003). The Beachhead approach can deliver the highest benefits but this approach is most difficult to implement (Simpson 2003).

Regardless of various platform formation strategies there are many techniques of implementing these strategies reported in the literature.

Various techniques of implementing platform strategies presented in literature include, but not limited to:

- Developing commonality matrices (Martin and Ishii 1997, Kota *et al.* 2000)
- Model based approach (Simpson *et al.* 1999, Farrell and Simpson 2001, Nayak *et al.* 2002)
- Analytical and mathematical approach (Lee and Tang 1997,

Swaminathan and Tayur, 1998)

- Optimization based approaches (Nelson *et al.* 2001, Gonzalez-Zugasti *et al.* 2000)

The research in this thesis falls in the category of the optimization based approaches for platform formation. Therefore, the next section provides a review on optimization based approaches for product formation problems.

2.1.4 Optimization based approaches for platform formation with various objectives

The platform design and selection concept have been used for various objectives. Such as, reducing cost and simplifying the design effort (Simpson 2004), improving life-cycle design (Ortega *et al.* 1999), optimizing production cost or profit, or reducing time to market (Krishnan and Ulrich 2001). Martin and Ishii (1997) proposed methodologies that can help companies to quantify the costs of providing variety and qualitatively guiding designers in developing products that incur minimum variety costs. Simpson *et al.* (1999) proposed a model that uses the overall design requirements, generating the product platform and resulting product family that best satisfies the overall design requirements. Farrell and Simpson (2001) try to improve response to customer request, reduce design cost and improve time to market for highly customized products by designing product platforms. Sudjitanto and Otto (2001) uses a matrix to group modules for platform determination in order to support multiple brands for platform cost saving as well as revenue enhancing. Nayak *et al.* (2002) proposed a variation-based method for product family design, which aims to satisfy the range of performance requirements for the whole product family.

Besides, the platform optimization problem appears in a variety of forms in the literature such as the product portfolio planning (Jiao and Zhang 2005) that finds an optimal set of products and attributes to satisfy customer choices and maximize expected utility per cost, or profit (Yano and Dobson 1998).

Most of the optimization approaches specify the platform a priori to the optimization to make the problem more tractable (Simpson 2003). Simpson and D'Souza (2002) encourage the use of optimization to explore varying levels of platform commonality for better platform development. The research in this thesis tries to optimize the platform configuration and platform based production simultaneously.

Various solution methods for the platform optimization problem were implemented including (but not limited to) Branch and Bound algorithm (Fujita and Yoshida 2001), Dynamic Programming (Allada and Jiang 2002), agent based techniques (Rai and Allada 2003), Simulated Annealing (Fujita *et al.* 1999), Genetic Algorithms (Fujita and Yoshida 2001; Li and Azaram, 2002, Simpson and D'Souza 2002, Simpson and D'Souza 2004, Jiao and Zhang 2005).

2.2 Genetic search approaches for large scale optimization

While small-dimensioned problems can be solved to optimality, large-scale problems require heuristic approaches such as Genetic Algorithms. During the last few decades, there has been a growing interest in evolutionary algorithms partially due to the emergence of faster computers. Genetic Algorithms have been studied in Goldberg (1989), Mitchell (1998), and a review of GA applications is provided in Aytug *et al.* (2003).

The solution methodologies presented in this thesis exploit the principles of evolution and mutation, and the concept of fitness. In general, the genetic search procedure starts with a random generation of population of strings (chromosomes), where each string represents a configuration (component set) of platform. The number of strings forming a population is termed population size, which remains constant throughout this genetic search process. The cost function, which should be minimized, is converted to a *fitness value*. A fitness function evaluates each solution to decide whether it will contribute to the next generation of solutions. The population then evolves through successive generations by the application of *genetic operators*.

The various genetic operators are *reproduction*, *cross-over* and *mutation*. The reproduction operator is an artificial version of natural selection, a Darwinian survival of the fittest among string creatures (Goldberg 1989). In the process of reproduction, strings having better fitness values have a higher probability of contributing one or more off-springs to the next generation. Reproduction directs the algorithm towards convergence.

In the crossover operator, genes (fractions of the strings) are swapped between two parent chromosomes in anticipation that the off-springs produced would be better than either of the parents. The operator contributes to the exploration of the solution space. The *mutation*, which creates a random variation in a string, reduces the chance of the algorithm converging to a local optimum. This approach uses the notation of $(\mu + \lambda) - ES$ (Beyer and Schwefel 2002), implying a constant population of size μ, with λ decedents created every iteration (for a total of μ+λ solutions) out of which the best μ are selected for the next generation. The algorithm continues until a pre-specified condition is met.

CHAPTER 3 - Problem Environment

3.1 Background

This section provides the underlying assumptions and some hypotheses on the basis of which three platform based production models (presented in Chapters 3, 4 and 5) are proposed and developed.

In this research, a platform is considered to be a set of shared components among multiple products. The platform is mass-produced, and a product from a product family is produced using a platform by adding or removing some of the components that are assembled using the platform. For instance, Figure 3.1 illustrates a hypothetical product family with four products (*P1, P2, P3,* and *P4*), each consisting of a different collection of components from the set {*A, B, C,…, H*}. Suppose a platform for this set of products is as shown in Figure 3.2. In this case *P1* would be created by using the platform and removing *G* and adding *C*, and *P3* would be created by removing *D* and *G* and adding *C* and *F*.

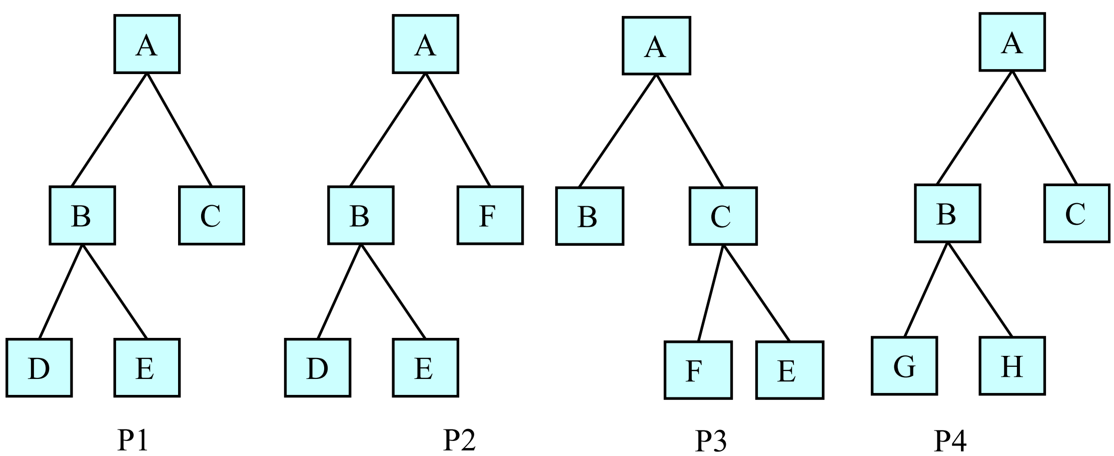


Figure 3.1 Example of a product family

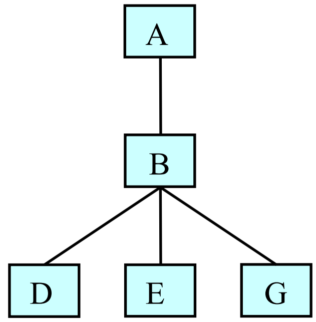


Figure 3.2 An example platform for the product family presented in Figure 3.1

A platform is only justified if the assembly of the components to the platform can be done efficiently using mass production methods. Thus, adding and removing components from a platform to fit a particular product typically costs more than if the item is included in the platform.

In this research, we model and analyze the production of a family of products using platforms that enable cost effective production with short final assembly time. The bill of material of each product is considered to be binary. A binary bill of material for the product family presented in Figure 3.1 is shown in Table 3.1.

Table 3.1 A binary bill of material for product family presented in Figure 3.1

Component index → Products	A	B	C	D	E	F	G	H
P1	1	1	1	1	1	0	0	0
P2	1	1	0	1	1	1	0	0
P3	1	0	1	0	1	1	0	0
P4	1	1	1	0	0	0	1	1

One complicating factor is that while determining the configuration of the platform (a set of components forming a platform), the part assembly relationship is maintained. The part assembly relationship for a product is presented by a matrix, and in order to manage the part assembly relationship constraints, a Part Assembly Relationship matrix for a product is determined. An element of Part Assembly Relationship matrix of product k, $f_{ljk}=1$, represents that component j precedes component l in product k or component j is needed to be present in the platform for the l to be included in the platform, as component l requires j to be assembled to form a platform. An example, a *PAR* matrix for product *P1* is shown in Table 3.2.

Table 3.2 The *PAR* matrix for Product *P1*

	A	B	C	D	E	F	G	H
A		1	1					
B				1	1			
C								
D								
E								
F								
G								
H								

As we will see, the *PAR* matrix is used to determine the feasibility of a platform configuration.

Also, for some platform based production models presented in this research the part assembly relationship matrix for the whole product family is used to determine the feasibility of a platform configuration. The part assembly relationship matrix for the whole product family, named as *Overall PAR*, is determined by taking all the *PARs* for all the products in the product family and superimposing them to get a superset type of matrix. For example, *Overall PAR* for the product family presented in Figure 3.1 would be as shown in Table 3.3.

Table 3.3 The *Overall PAR* matrix for Product family presented in Figure 3.1

	A	B	C	D	E	F	G	H
A		1	1			1		
B				1	1		1	1
C					1	1		
D								
E								
F								
G								
H								

The PAR matrix for a product and the Overall PAR matrix for a product family are used to check and the feasibility of a given platform configuration in different platform based production model presented in this thesis.

3.2 Example Problem

This section provides a description of an example problem that will be used to illustrate and numerically validate the proposed models and solution approaches proposed in this thesis.

We use an example of a family of cordless drills. These drills have varied construction and household applications. There are various types of drills available; however, for our purpose we consider five types (products): Heavy Duty, High Performance, High Value, Standard, and Multipurpose Power drills as shown in Figure 3.3 (Sudjiato and Otto, 2001). The objective in this example is to demonstrate the use of the heuristics to determine the optimal platform for the product family. The binary bill of material for family of drills is provided in Table 3.4. The information about the products and the components is provided in Table 3.5. The Table 3.6 provides the PAR matrix for "Heavy Duty" drill, for the sake of brevity the PAR matrix for all the drills are not presented. Table 3.7 provides the Overall PAR for the whole drill family.

High Value	Heavy Duty	High Performance	Standard	Multipurpose

Figure 3.3 The product family of the cordless drills

Table 3.4 The binary bill of material for the products

Component # →	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Component index → Products	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Heavy duty	1	1	0	0	1	0	1	0	1	1	1	1	0	1	0	0	0	1	0	1	0	1	0
High performance	1	0	1	0	0	1	0	1	1	1	1	0	1	0	1	0	0	0	1	0	1	1	0
High value	1	0	0	1	0	1	0	1	1	1	1	0	0	0	0	1	0	0	0	0	1	1	0
Standard	1	0	0	1	0	1	0	1	1	1	1	0	0	0	0	1	0	0	0	0	1	0	1
Multi-purpose	1	0	0	1	0	1	0	1	1	1	1	0	0	0	0	0	1	0	0	0	1	0	1

Table 3.5 The various possible components for the product family with there cost values

Comp #	Comp. Index	Comp. Name	Comp. Cost ($)
1	A	Encasing	2
2	B	Rough palm to permit positioning	1
3	C	Padded palm to permit positioning	1.5
4	D	Diamond palm to permit positioning	2
5	E	Bevel 2 point to lock/unlock battery	3
6	F	Straight 2 point to lock/unlock r battery	2
7	G	Square, 9.6 V, 2 pt. Electricity transmission	4
8	H	Open, 9.6 V, 2 pt. Electricity transmission	4
9	I	Chuck to secure/ unsecure bit	2
10	J	Chuck teeth to register/ unregister bit	1
11	K	Bit to act on object	2
12	L	Thin button to input speed	1
13	M	Wide button to input speed	1
14	N	16 slip clutch to transmit power	5
15	O	22 slip clutch to transmit power	6
16	P	6 slip clutch to transmit power	5
17	Q	Solid shaft to transmit power	3
18	R	Fine ring gear to switch speed	5
19	S	Ring gear to switch speed	4
20	T	Black oval button to unlock switch	1
21	U	Black button to unlock switch	1
22	V	Variable speed to switch power	6
23	W	2 speed to switch power	4

Table 3.6 The PAR matrix for the "Heavy Duty" drill

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
A		1			1				1														
B																							
C																							
D																							
E							1											1					
F																							
G								1															
H																							
I										1													
J											1	1											
K														1									
L																							
M																							
N																							
O																							
P																							
Q																				1		1	
R																							
S																							
T																							
U																							
V																							
W																							

Table 3.7 The Overall PAR matrix for the family of cordless drills

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
A		1	1	1		1		1															
B				1	1	1				1													
C						1			1		1										1		
D									1	1						1							
E						1	1		1	1	1												
F								1			1												
G								1															
H									1	1													
I										1	1												
J											1												
K												1									1		
L													1										
M														1									
N															1								
O																1							
P																	1						
Q																		1				1	
R																			1				
S																				1			
T																					1		
U																						1	1
V																							1
W																							

3.3 Notations and Nomenclatures

This section provides the notations and nomenclatures used throughout the thesis.

- $i = 1, 2, \ldots, |I|$ index the platforms, where I is a set of platform types
- $j,\ l$ = 1, 2, …, M index the components, where M = total number of all distinct components in a given product family, and $|J|$ represents the set of components
- k = 1, 2… N index the products, where N is total of products in the given product family, $|K|$ represents the set of products.

- $s = 1, 2, \ldots, S$ index the demand scenarios considered in a stochastic demand model
- $D_k =$ demand of the k^{th} product. This demand is usually forecasted, since precise demand may not be known with certainty at that decision point.
- C_j= cost of the component j (purchasing price)
- CP_j= cost of assembling a component j to form the platform (mass assembly)
- CA_j= cost of manually adding a component to the platform to form a product $(CA_j > CP_j)$
- CR_j= cost of removing a component from the platform to form a product $(CR_j > CP_j)$
- A_i = the setup cost to construct platform i
- h = per unit holding cost for the platforms
- q_k= per unit stock-out cost for product k
- ξ_s = Vector of demands $(\xi_{1s}, \xi_{1s}, \ldots, \xi_{Ns})$ in scenario s
- p_s = probability of occurrence of scenario s
- V is the given binary bill of material matrix of the family of products with element $v_{jk} = \begin{cases} 1 \text{ if product } k \text{ requires component } j \\ 0 \text{ otherwise} \end{cases}$
- f_{jlk} are elements in the Part Assembly Relationship matrix of product k with $f_{jlk} = \begin{cases} 1 \text{ if component } j \text{ precedes } l \text{ in product } k \\ 0 \text{ otherwise} \end{cases}$
- $f_{jl} = \begin{cases} 1 & \text{if component } j \text{ precedes component } l \text{ according to Overall part assembly} \\ & \text{relationship matrix of the given product family} \\ 0 & \text{otherwise} \end{cases}$
- X = a binary matrix representing which platform contains which components with elements, $x_{ij} = \begin{cases} 1 \text{ if platform } i \text{ contains component } j \\ 0 \text{ otherwise} \end{cases}$
- $x_j = \begin{cases} 1 & \text{if component } j \text{ becomes the part of the platform} \\ 0 & \text{otherwise} \end{cases}$
- w = quantity of the platforms to be made
- Y is a binary matrix that states that product k is made on platform i, with elements

$$y_{ki} = \begin{cases} 1 \text{ if product } k \text{ is made using platform } i \\ 0 \text{ otherwise} \end{cases}$$

- y_{ks} = amount of product k to be produced using platforms in scenario s
- $a_{ijk} = \begin{cases} 1 \text{ if the } j^{th} \text{ component is added manually to platform } i \text{ to form product } k \\ 0 \text{ otherwise} \end{cases}$
- $a_{jk} = \begin{cases} 1 & \text{if component } j \text{ is added manually to the platform to make product } k \\ 0 & \text{otherwise} \end{cases}$
- $r_{ijk} = \begin{cases} 1 \text{ if the } j^{th} \text{ component is removed manually from platform } i \text{ to form product } k \\ 0 \text{ otherwise} \end{cases}$
- $r_{jk} = \begin{cases} 1 & \text{if component } j \text{ is removed manually from the platform to make product } k \\ 0 & \text{otherwise} \end{cases}$
- u_{ks}^{-} = Lost demand of product k in scenario s
- v_s^{+} = Leftover platforms in scenario s
- *PAR* = Part Assembly Relationship Matrix for a Product
- *Overall PAR* = Overall Part Assembly Relationship Matrix for all the products in the given product family
- p_s = Crossover Probability
- p_m = Mutation Probability

CHAPTER 4 - Economic Production of a Product Family using Single Platform

4.1 Introduction

In this chapter, we model and analyze the production of a family of products based on single platform which enables cost effective production with short final assembly time. We consider a problem of selecting a platform for a product family while minimizing the overall production cost which includes cost of components, cost of mass assembly, and cost of adding/ removing components to the platform. The problem is formulated as a general optimization problem.

Both an optimal and an evolutionary strategy based on Genetic Algorithm are proposed for the problem. The approaches are illustrated with an example of the family of cordless drills. The example is used to provide insights to the effect of demand variance and various cost components on the optimal configuration of the platform. Finally, we discuss the effectiveness of the heuristic tailored for the application.

4.2 Model Formulation

The problem is to determine the optimal configuration of the platform for a given product family to minimize the total production costs. Every product, *k, (1…, k…, N)* may either be assembled directly from its components, or from any platform whose component set overlap with those required by product *k*.

The bill of material of the product family in terms of components is binary. While determining the optimal configuration of the platform, the part family relationship is maintained.

Now, the optimal platform configuration determination problem can be formulated as follows:

Minimize

$$\sum_{k=1}^{N} D_k \times \sum_{j=1}^{M} \left(C_j + CP_j\right) \cdot x_j + \sum_{k=1}^{N}\sum_{j=1}^{M} \left(C_j + CA_j\right) \cdot a_{jk} \cdot D_k + \sum_{k=1}^{N}\sum_{j=1}^{M} \left(CR_j - C_j\right) \cdot r_{jk} \cdot D_k \qquad (4.1)$$

S.t.

$$a_{jk} = (1 - x_j) \cdot v_{jk} \qquad \forall j,k \qquad (4.2)$$

$$r_{jk} \leq (1 - v_{jk}) \cdot x_j \qquad \forall j,k \qquad (4.3)$$

$$x_j, r_{jk}, a_{jk} \in \{0,1\} \qquad \forall j,k \qquad (4.4)$$

The objective (equation 4.1) is to minimize the total production cost, which includes the cost of mass assembly (cost of producing platforms), cost of components, cost of adding components on the platform to produce the products, and the cost of removing components from the platforms. Constraint 4.2 ensures that a component is added to the platform only if it is required in the product and is not in the platform. Constraint 4.3 ensures that a component may be removed from the platform only if it is in the platform and not required in the product.

Example:

We illustrate the problem through an example. We use OPL 3.5 studio to find the optimal solution to the above problem with various demand and cost data. The optimal platforms for the various cases are shown in Table 4.1.

The various components costs are $C_j s$= \$10, \$11, \$12,…., \$17; CR_j=\$ 9 (for all j); CP_j = \$6 (for all j); and CA_j= \$10 (for all j).

Table 4.1 The optimal solution for the various cases

Demands				**Optimal Platform**	**Total Cost (\$)**	**Time (Seconds)**
D1	**D2**	**D3**	**D4**			
250	250	250	250	AB	104550	58
350	150	250	250	AB	104550	72
250	450	150	150	AB	104550	61
800	50	50	100	ABCE	99700	56

From Table 4.1, we can see that when the demands of the products are similar and that the optimal platform comprises the components are the most common across the product family. Also, when the demand of a product is very high when compared to other products in the product family, the optimal platform configuration is more inclined towards the product with high demand.

4.2 The Evolutionary Solution Methodology

We propose a very efficient evolutionary search based solution methodology for the platform determination problem. The algorithm presented uses a variation mechanism that is derived by inducing mutations to the entire (100%) population. This evolutionary strategy provides sufficient exploration of the solution space, needed for a successful solution.

The algorithm starts with a small number of randomly generated solutions. These solutions are tested for feasibility, and all infeasible solutions are modified to become feasible, as shown in Section 4.3.1.

The algorithm tries to improve the feasible solutions using the evolutionary methodology. The algorithm is terminated after a predefined number of iteration is reached, and the best solution is anticipated to be the near optimal/optimal solution. The detailed explanation of the algorithm is provided next, with a flow chart of the algorithm provided in Figure 4.1.

In the platform determination problem, we are given the binary bill of materials for all the products in terms of components. Also, the part assembly relationship of the products is known from the Part Assembly Relationship Matrix. A solution consists of the values of the binary decision variable x_j and the variables r_{jk}, indicating the parts removed from the platform. The solution is represented by a string of length equal maximum number of distinct components in the product family under question, *M,* and a matrix representing r_{jk}. An entry of '1' at any position *j* in the string x_j represents component *j* that is in the platform. Each solution (or chromosome) in the population is a feasible solution, which follows the part assembly relationship for the product family under consideration. The lower the cost of a particular solution, the higher the fitness value it possesses. The details of the algorithm are provided next.

4.3.1 Encoding and Initial Feasible Solution Generation

Solution Encoding

In the platform determination problem, we are given the binary bill of materials (v_{jl}) for all the products. Also, the part assembly relationship of the products is provided in PAR Matrix. For both the heuristic approaches, the chromosome is represented by a string of length equal to the maximum number of distinct components in the product family considered, *M.* A value of '1'at any position *j* in the string represents component *j* that is in the platform. An example chromosome string for the product family shown is Figure 3.1 is presented below.

A	B	C	D	E	F	G	H
1	1	1	0	0	0	0	0

This example chromosome string represents that components A, B, and C are in the platform.

Each chromosome (or configuration of a platform which is the component set of a platform) in the population is feasible if it follows the part assembly relationship for the product family under consideration (such as in Table 3.2). Initially a population of chromosome is randomly generated; therefore some of the solutions may not be feasible. Also, applying the variation operator (mutation) create new temporary solution that may not be feasible.

If the solution is found infeasible it has to be corrected. Every time the chromosome is altered in the course of iterations, it has to be checked for feasibility and corrected if not feasible. Therefore, for this purpose a feasibility and correction algorithm is presented as follows.

Feasibility and correction algorithm

The idea behind the algorithm is that the feasibility of the solution is checked using the Overall Part Assembly Relationship Matrix and if a platform is infeasible, the smallest numbers of components are added to the platform to ensure a feasible solution.

Let *P* (a binary string) represents a random solution string. This solution may be feasible or infeasible. Let T_j (a binary string) represents the j^{th} column in the Overall PAR Matrix. For example from Table 3.2, T_5 (or T_E) = [0 1 1 0 0 0 0 0].

The Algorithm:

Step # 1: Determine all strings Q_j = *P* ***AND*** T_j for all j=1 in *P*

Step # 2: Compare Q_j with T_j bitwise

Step # 3: If any Q_j not equals T_j the solution is infeasible, go to step 4; otherwise feasible go to step 5.

Step # 4: Create a new feasible *P'* = [bitwise *OR* (T_j)] ***OR*** *P* for all j=1 in *P*, go to step 2. (where *P'* is the corresponding feasible solution for any infeasible solution *P*).

Step 5: Stop

Example:

Let P be a randomly generated platform for the product family in Figure 3.1.

P= [1, 0, 0, 0, 1, 1, 0, 0]

Now from Table 3.2, T_1= [0, 0, 0, 0, 0, 0, 0, 0]

$Q_1 = (P \text{ } AND \text{ } T_1)$= [0, 0, 0, 0, 0, 0, 0, 0]

Also:

T_5= [0, 1, 1, 0, 0, 0, 0, 0]

Thus, Q_5 = [0, 0, 0, 0, 0, 0, 0, 0]

T_6 = [1, 0, 1, 0, 0, 0, 0, 0]

And Q_6 = [1, 0, 0, 0, 0, 0, 0, 0]

As the platform generated is infeasible and the solution is made feasible by taking P' = [bitwise ***OR*** (T_j's)] ***OR*** P

For this example, [bitwise ***OR*** (T_1, T_5, T_6)] = [1, 1, 1, 0, 0, 0, 0, 0].

And the feasible platform is:

P' = [1, 1, 1, 0, 1, 1, 0, 0], which coincidentally represents Product, P3 presented in Figure 3.1.

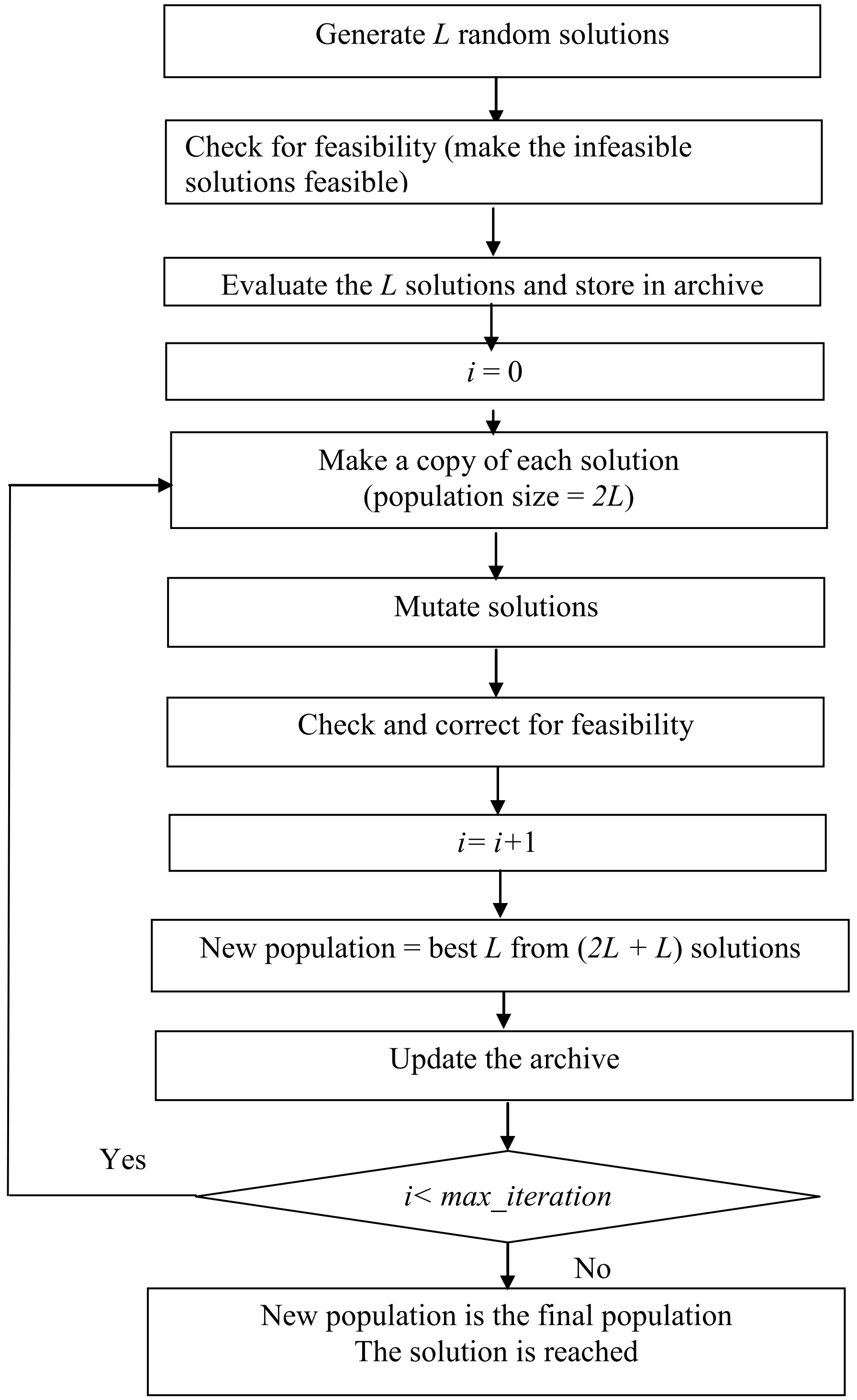


Figure 4.1 The flow chart of the evolutionary strategy

4.3.2 Mutation

Given a solution *S*, a new solution *S'* = mutation (*S*) can be built using a mutation operator. In our case, we use a random mutation in the following way:

Generate two ransom numbers *j* and *l* such that $0 < j < l < M$. Invert the bits in positions *j* and *l* only.

The process can be represented as follows (Figure 4.2).

S=	1	1	1	0	0	0	0	0
				⇩				
S' =	1	1	0	0	1	0	0	0

Figure 4.2 Mutation Operation

Next check the feasibility of *S*'. If it is not feasible, modify the string to become feasible as shown in Section 4.3.1.

4.3.3 Building Each Generation

The algorithm starts with the random generation of *L*=10 solutions. The feasibility of each solution is checked and if not feasible a corresponding feasible solution is generated as explained above. These solutions are evaluated and kept in an archive. Then these solutions and a copy of each of them are taken for mutation. After the mutation step we have *2L* new solutions. These solutions are checked and corrected for feasibility. These mutated feasible solutions are evaluated and the best *L* of the *L+2L* solutions updates the archive (new initial feasible population).

The population size is doubled during the phase mutation to make the search more exploratory. Several other mutation techniques are investigated but the above mentioned mutation technique works well for the problem under consideration.

4.3.4 Solution Evaluation

Each suggested platform has a cost function that shows the cost of converting the platform to each of the products, given each product demand (production quantity). The cost calculations follow this algorithm:

For each component j in the platform Do

$Cost = (C_p+C_j)*\Sigma D_k$

For each product k Do:

If j is NOT in product k Then

If $CR_j > C_j$ *Then*

$r_{jk} = 0$

$Cost = Cost$

Else

$Cost = Cost + CR_j*D_k$

End k

End j

For each component j Not in platform Do

For each product k Do:

If j is in the product Then

$a_{jk} = 1$

$Cost = Cost + D_k*(CA_j + C_j)$

EndIf

End k

End $_j$

The algorithm is applied to the example problem presented in Section 3.2 and the results with the computational insights are presented in the following Section.

4.4 Results and Discussions

A convergence plot for the case of demand vectors of [50, 250, 250, 250, 200] (for drills), and costs, CP_j, CA_j, and CR_j equals \$1, \$4 and \$3 respectively for all *j* and *k*, is shown in Figure 4.3. It is obvious that the algorithm converges efficiently with relatively less iterations. Also, the r_{jk} matrix for that case is provided in Table 4.2.

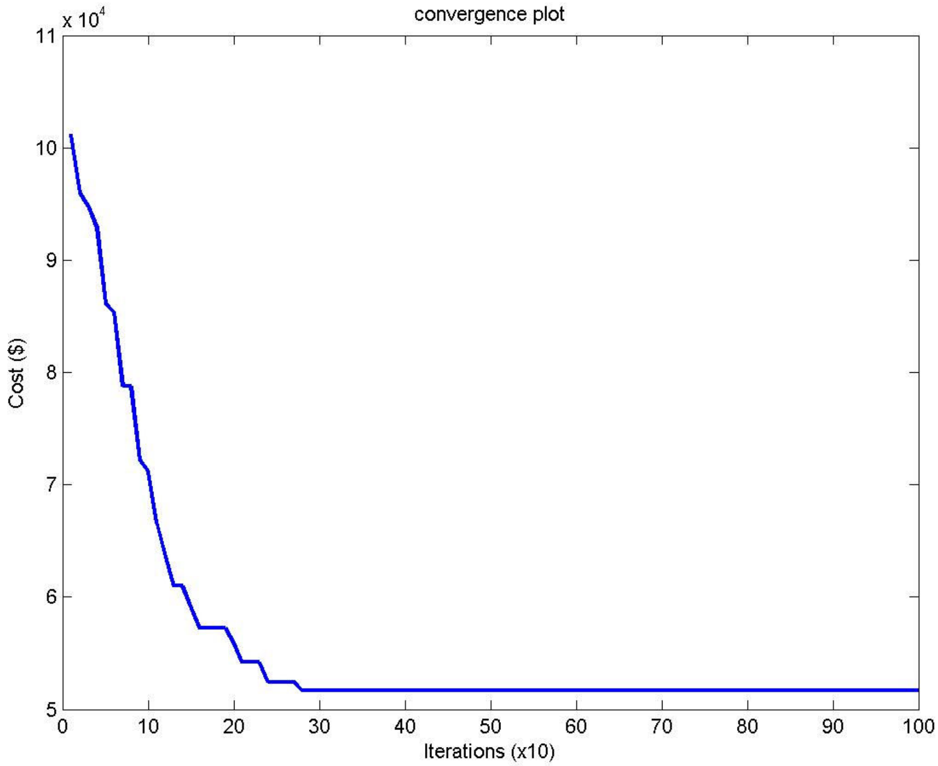


Figure 4.3 The convergence plot of the algorithm

Table 4.2 The r_{jk} Matrix

Component # → Product	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Heavy duty								**1**								**1**							
High performance																**1**							
High value																							
Standard																						**1**	
Multi-purpose																**1**						**1**	

For further investigation we determine the optimal platform for various cases of demand and CP_j, CA_j and CR_j values. The results are summarized in Table 4.3.

Table 4.3 The platform and overall cost for various cases of demand and cost values

Demands	Costs (CP_j, CA_j and CR_j)	Platform	Cost value ($)
200, 200, 200, 200, 200	2, 4, 3	1 6 9 10 11	62500
50, 250, 250, 250, 200		1 4 6 8 9 10 11 15 20	64425
250, 50, 250, 200, 250		1 4 6 9 10 11	61075
200, 250, 250, 50, 250		1 6 9 10 11	63275
250, 250, 200, 250, 50		1 6 9 10 11	64675
200, 200, 200, 200, 200	3.25, 3, 2	-	57500
50, 250, 250, 250, 200		-	55425
250, 50, 250, 200, 250		-	56675
250, 200, 50, 250, 250		-	58300
200, 200, 200, 200, 200	1, 4, 3	1 4 6 8 9 10 11	56100
50, 250, 250, 250, 200		1 4 6 8 9 10 11 16 22	51675
250, 50, 250, 200, 250		1 4 6 8 9 10 11 16 22	56675
200, 250, 250, 50, 250		1 4 6 8 9 10 11 17 21	59675
250, 250, 200, 250, 50		1 6 8 9 10 11 17 21	64175
900, 25, 25, 25, 25		1 2 5 7 9 10 11 12 14 18 20 22	50438

Note: The demand vector represents the following order of products: Heavy Duty, High Performance, High Value, Standard and Multi Purpose (*P1, P2, P3, P4,* and *P5*)

The following results are obtained;

1. From Table 3.4 and Table 4.3, it is obvious that when the demand of each product is similar enough, platform elements are the elements that are the most common throughout the product line, regardless of the components cost.
2. When the cost of mass assembly of each component in the platform exceeds or equals the cost of manually adding the component to the platform, the production of platform should not be justified. The results found in Table 4.3 support this conclusion.
3. When the demand of a particular product is very high with respect to others in the product line, the platform components are those which are in that product, even if these components are not shared by other products. For example, if the demand of Heavy Duty drill is 900 and demand for the rest is equal 25 each, from Table 4.3 (and Table 3.4) we

can see that the platform, in this case, is the whole Heavy Duty type drill itself. These observations clearly show that the results obtained from the approach are consistent.

Comparison with the Optimal Algorithm:

Now we describe experimental results that compare the evolutionary algorithm to the optimal solution obtained by solving the integer programming model. In this experiment, 6 different variants of the example problem with 23 components were solved. The OPL 3.5 takes about 160 minutes to solve the example problem one time, whereas the heuristic takes less than a minute to solve the problem one time. Each problem was solved 20 different times using the heuristic approach by taking different initial solution (platform configuration) each time and the best solution out of 20 runs is recorded, the average of the solutions obtained from 20 runs is determined as well. Table 4.4 presents the results of the analysis in the following way. The first column presents the parameters of the problem solved, followed by the optimal cost. The next four columns represent the results of the evolutionary approach, starting with the best solution achieved (from the 20 solved), followed by the percent difference between the optimal and the best solution. Then, the average solution is presented with its percent deviation from the best solution. From the table it is seen that the evolutionary approach found the optimal solution in two problems (under the specific conditions of those problems were no platform was recommended). The rest of the four problems show that the heuristic approach (evolutionary) reached a very good solution – within about 1 percent in most cases.

Table 4.4 Computational Analysis of Evolutionary Approach

Problem		Optimal solution	Evolutionary Approach			
#	Demand [P_1, P_2, P_3, P_4, P_5], Cost (CP_j, CA_j and CR_j) in dollars		Best	%Difference (Best vs. optimal)	Average	%Difference (Best vs. Average)
1	[200, 200, 200, 200, 200] (2, 4, 3)	59900	62500	4.34	63025	0.8
2	[50, 250, 250, 250, 200] (2, 4, 3)	58275	58425	0.257	59500	1.8
3	[200, 200, 200, 200, 200] (3.25, 3, 2)	57500	57500	0	57500	0
4	[50, 250, 250, 250, 200] (3.25, 3, 2)	55425	55245	0	55425	0
5	[200, 200, 200, 200, 200] (1, 4, 3)	55400	56100	1.26	57200	3.24
6	[50, 250, 250, 250, 200] (1, 4, 3)	51225	51425	0.39	51500	0.1

4.5 Conclusion

This chapter presents the concept of a common platform as a solution to the production of a family of products in a cost effective manner. The chapter presents a description of the problem followed by a mixed integer formulation presented as an optimization problem. Then an evolutionary strategy based on Genetic Algorithm is proposed for the problem. The approach is explained and illustrated using an example of a family of cordless drills. The heuristic approach is found to very fast when compared to the exact approach, provided solution within 1% error for most of problem instances. Also, the chapter provides insight into the effects of demand variance and various cost components on the optimal configuration of the platform.

CHAPTER 5 - Economic Production of a Product Family using Multiple Platforms

5.1 Introduction

In this chapter, we present a methodology for selecting multiple platforms for the production of a product family. The advantage for using multiple platforms over a single platform is the ability to optimally match products to a particular platform. Most of the product platform formation formulations consider only a single platform analysis.

The chapter presents two solution approaches to the problem – an exact solution and a heuristic approach. The problem is solved exactly as an MIP problem where the constraints are made linear and some valid cutting planes are suggested. Even with adding the cutting planes, the MIP can require substantial computation time. Therefore, a genetic algorithm is presented that can quickly provide good solutions even to large instances. Both of these approaches are illustrated with a small numerical example and a larger example of a product family of Cordless drills. These examples are used to provide insights to the effect of demand variance and various cost components on the optimal configuration of the platforms. Also, the heuristic's solution quality is discussed by comparing the two approaches.

The contribution of this chapter is in introducing the problem of multiple platforms, and providing an efficient mixed integer program. Moreover, the genetic algorithm is also unique since it is searching for an unknown number of solutions corresponding to the platforms adopted for the production.

5.2 Model Formulation

In this section we model and analyze the production of a family of products using multiple platforms that enables cost effective production with short final assembly time. This model enables the systematic determination of the optimal number of product platforms, the configuration of each platform (its component set) and the assignment of each platform to the various products in the family, while minimizing the overall product family production costs.

The optimal multiple-platform configuration determination problem can be formulated as:

$$
\begin{aligned}
Min \quad & \sum_{i\in I}\sum_{j\in J}\sum_{k\in K}\left(CP_j + C_j\right)\cdot x_{ij}\cdot y_{ki}\cdot D_k + \\
& \sum_{i\in I}\sum_{j\in J}\sum_{k\in K}\left(CA_j + C_j\right)\cdot a_{ijk}\cdot y_{ki}\cdot D_k + \\
& \sum_{i\in I}\sum_{j\in J}\sum_{k\in K}\left(CR_j - C_j\right)\cdot r_{ijk}\cdot y_{ki}\cdot D_k + \\
& \sum_{i\in I} A_i
\end{aligned}
\tag{5.1}
$$

Subject to:

$$a_{ijk} = \left(1 - x_{ij}\right)\cdot v_{jk}\cdot y_{ki} \qquad \forall\ i \in I;\ j \in J;\ k \in K \tag{5.2}$$

$$r_{ijk} \le \left(1 - v_{jk}\right)\cdot x_{ij}\cdot y_{ki} \qquad \forall\ i \in I;\ j \in I;\ k \in K \tag{5.3}$$

$$\sum_{i=1}^{|I|} y_{ki} = 1, \qquad \forall\ k \in K \tag{5.4}$$

$$x_{ij} \ge f_{jlk}\cdot y_{ki}\cdot x_{il} \qquad \forall i \in I; j,l \in J; k \in K \tag{5.5}$$

$$I \in \{1,\ 2, \ldots,\ K\} \tag{5.6}$$

$$x_{ij} \in \{0,\ 1\};\ y_{ki} \in \{0,\ 1\};\ a_{ijk} \in \{0,\ 1\};\ r_{ijk} \in \{0,\ 1\} \tag{5.7}$$

Decision Variables: x_{ij}, y_{ki}, a_{ijk}, r_{ijk} The objective minimizes the cost, which includes the setup costs for each platform, the optimal set of components to include in each platform, and the optimal assignment of products to platforms. The first term in the objective function (Equation 5.1) represents the cost of production of platforms, second term represents the cost of adding components manually to the various platforms to form different products, the third term represents the cost of manually removing (and allowing for reuse) excessive components from the platforms to form each product, and the final term represents the setup cost of constructing the platforms.

Constraints (5.2) restrict component *j* to be added to platform *i* to make product *k* only if the component is not already in that platform. Thus, component *j* is required for product *k*, and product *k* is assigned to platform *i*. Constraints (5.3) state that a component *j* may be removed from platform *i* if that component is not required in product *k*, the component is assigned to platform *i* and product *k* is assigned to that platform. Constraints (5.4) ensure that each product is made by only one platform. Constraints (5.5) check the assembly feasibility of each product that uses a platform so that if component l precedes component j in a product k assigned to the platform, and component l is assigned to the platform, then component j must be on the platform. Constraints (5.6) represent that the optimal number of platforms is an integer and the maximum number of platforms is limited by the total number of the products in the family, and Constraints (5.7) ensure binary decision variables.

5.2.1 Improving the Formulation

In the formulation constraints (5.2), (5.3) and (5.5) are non-linear, which makes selecting a solution procedure difficult at best. The following changes are made to constraints (5.2), (5.3) and (5.5), which make the formulation linear.

$$a_{ijk} + x_{ij} \le 1 \qquad \forall i \in I; j \in J; k \in K \qquad (5.8)$$

$$a_{ijk} + x_{ij} \ge v_{jk} \qquad \forall i \in I; j \in J; k \in K \qquad (5.9)$$

$$v_{jk} \ge a_{ijk} \qquad \forall i \in I; j \in J; k \in K \qquad (5.10)$$

$$x_{ij} \ge r_{ijk} \qquad \forall i \in I; j \in J; k \in K \qquad (5.11)$$

$$r_{ijk} + x_{ij} + v_{jk} \le 2 \qquad \forall i \in I; j \in J; k \in K \qquad (5.12)$$

$$1 + x_{ij} \ge f_{jlk} \cdot y_{ik} + x_{il} \qquad \forall i \in I; j, l \in J; k \in K \qquad (5.13)$$

Equations (5.8)-(5.10) replace the non-linear constraints (5.2), equations (5.11)-(5.12) replace non-linear constraints (5.3) and equation (5.13) replaces the non-linear constraints (5.5). The solution space is extremely large. For instance, the total possible platform configuration being $2^{|J||I|}$, which is just one decision variable of the model.

To help reduce the search space we introduce some cutting planes. The first cut was added to avoid the symmetrical nature of the problem. In this case the same solution can be represented in $|I|$ different ways by merely permuting the platforms. To eliminate symmetry the following additional constraints are used:

$$\sum_j x_{ij} \ge \sum_j x_{sj} \quad \forall i, s \in I \qquad (5.14)$$

$$\sum_k y_{ki} \ge \sum_k y_{ki} \quad \forall i, s \in I \qquad (5.15)$$

Another constraint that prevents the same component from being added and removed from the same platform:

$$a_{ijk} + r_{ijk} \leq 1 \qquad \forall i \in I; j \in J; k \in K \qquad (5.16)$$

These cuts are included in the formulation and the model is solved. Adding cut (5.14) reduces the computational time by more than 50%, while cuts (5.15) and (5.16) had a smaller contribution.

5.2.2 An Illustrative Example

In this subsection a small example to illustrate the solution of the integer program is presented. This example uses a family of four products having eight distinct components, as shown in Figure 3.1. The cost of the components is given as

Component →	A	B	C	D	E	F	G	H
Cost ($)→	10	11	12	13	14	15	16	17

This example is solved optimally, and the results for the various cases of demand and costs are shown in Table 5.1. The results presented in Table 5.1 show that utilizing multiple platforms is economically justified. And this fact is even more pronounced when the setup cost of new platforms is relatively small.

The following conclusions can be drawn from the results:

1. For some values of setup cost, proposing more than one platform is cost effective.
2. With a decrease in setup cost of the platforms, the optimal number of platforms increases.
3. From Figure 3.1 and Table 5.1, it is evident that when the demand of each product is similar enough, platform elements are the elements that are the most common throughout the product line, independent of the components cost.
4. When the cost of mass assembly of each component in the platform exceeds the cost of manually adding the components to the platform, using platforms is not justified.

Table 5.1 The platform and overall cost for various cases of demand, and cost

Setup cost ($) (A)	Costs (CP_j CA_j CR_j)	Demand	Single platform	Cost in $ (Single)	Multiple platforms	Cost in $ (Multiple)	Solution Time (sec.)
1000	(2 4 3)	[250 250 250 250]	[AB]*	79750	[AB] (2, 4)** [ABCE] (1,3)	78750	128
		[700 100 100 100]	[ABCE]	77500	[AB] (2,3,4) [ABCDE] (1)	74900	130
		[100 700 100 100]	[AB]	79900	[ABD] (2) [ABC] (1, 3, 4)	78900	115
		[100 100 700 100]	[A BCE]	78700	[AB] (1, 2, 4) [ABCEF] (3)	76100	132
		[25 25 25 925]	[ABCGH]	80150	[AB] (1, 2, 3) [ABCGH] (4)	78125	126
	(2 1.75 3)	[250 250 250 250]	[-] ***	72500	[-]	72500	12
		[700 100 100 100]	[-]	72500	[-]	72500	12
100	(2 4 3)	[250 250 250 250]	[AB]	78850	[-] (2) [ABCGH] (4) [ABCE] (1,3)	76550	612
		[700 100 100 100]	[ABCE]	76600	[-] (-) [AB] (2, 3, 4) [ABCDE] (1)	74000	608
		[25 25 25 925]	[ABCGH]	79250	[-] (2) [ABCGH] (4) [ABCE] (1,3)	76325	598

* '[ ]' Represents the components set of the single platform and all the products in the product family is assigned to it.
** '[ ] ( )' Represents the components set of a platform and '( )' represents the products set out of the product family made from that platform
*** '[-]' Represents the platform doesn't have any component in it

5. When the demand of a particular product is very high with respect to others in the product line, the platform components are those that are in that product, even if other products do not share these components. For example, if the demand of P4 (Figure 3.1) is very high with respect to others in the family, the platform in this case is the product P4 itself.

5.3 Genetic Algorithm Solution Methodology

Observe that even the very small problem solved in Section 5.2 required around 10 minutes to solve. The solution time increases exponentially as the larger problems were attempted, and the solution was not reached even in a week. Therefore, we present a genetic algorithm (GA) based heuristic approach for the multiple-platform problem selection problem.

The algorithm starts by applying the genetic algorithm to a single platform model. Once the near optimal solution for the model is obtained, the genetic algorithm is applied by considering the number of optimal platform equals two for the model, and a near optimal solution is obtained. This continues until $|K|$ platforms have been analyzed. The smallest cost solution for k number of platforms is then reported as the solution to the multiple-platform problem selection problem and $|K|$ is the optimal number of platforms.

The genetic algorithm follows the steps presented in Section 5.3.1. The lower the cost-value of a particular chromosome, the higher the fitness value it possesses. The algorithm terminates after a pre-specified number of iterations and the best solution (with minimum cost value) is reported.

5.3.1 The Genetic Algorithm

Input: μ = the population size

p_c = the crossover probability

p_m = the mutation probability

T = the maximum number of generations (number of iterations)

P_t = the population on the t^{th} iteration

Step # 1: **Initialization**

Generate a random initial population P_0 of size μ and a random initial offspring population P'_0 of size μ, and apply feasibility and correction algorithm (described in Section 4.3.1) to create P_0 and P'_0 and set t=0.

Step # 2: **Fitness**

$P'_{t+1} \leftarrow P_t \cup P'_t$; and

Calculate the fitness of each individual in P'_{t+1}. The fitness, f_p, of an individual p is given by $f_p = 1/\cos t\ value$. Where *value cost* is the value of objective function for that individual (solution).

Step # 3: **Evolution**

Sort P'_{t+1} based on fitness value (in decreasing order) and truncate the size of P'_{t+1} to the best μ solutions. $P_{t+1} \leftarrow P'_{t+1}$.

Step # 4: Selection

Individuals from P_{t+1} are selected for mating. The individuals, of the same number as the population size, will be copied into a mating pool according to their fitness values. The higher the fitness values the greater the probability of individuals to join the mating pool (individuals may be selected for the mating pool more than one time.). The selection procedure is done as follows.

Begin

$p \leftarrow 0$;

While $(p \leq \mu)$ **do**

Calculate the selection probability and cumulative probability for an individual p as $\Pr_p = f_p \big/ \sum_p f_p, \forall p = 1, 2, ..., N$ and $Cr_p = \sum_{k=1}^{p} \Pr_k, \forall p = 1, 2, ..., N$ respectively; and generate a random number r, $r \in [0,1]$;

If $r \leq Cr_1$, **then** select the first individual; **else**, select the p^{th} individual $Cr_{p-1} < r \leq Cr_p, \ \forall p = 2, 3, ..., N$;

End if

$p \leftarrow p+1$;

End

Step # 5: **Variation**

Apply crossover with probability p_c and mutation with probability p_m to P_{t+1} to generate P'_{t+1}. With crossover probability p_c we mean that on an average $(p_c * \mu)$ individuals would undergo crossover to generate $(p_c * \mu)$ children. Then the mutation operator is applied with a low mutation probability p_m on P_{t+1}. After applying the crossover and mutation operators, we get on an average $(p_c + p_m) * \mu$ number of children, denoted as P'_{t+1}.

Step # 6: **Feasibility and correction**

Apply feasibility and correction algorithm to P'_{t+1} to create P_{t+1} and $t \leftarrow t+1$.

Step # 7: **Termination**

If $t = T$, terminate and print the solutions in P_t, otherwise go to Step #2.

5.3.2 Solution Representation

The solution is represented using a matrix with $|I|$ rows and $|J|$ columns, where $|J|$ is the maximum number of distinct components in the product family under consideration, as illustrated in Figure 5.1. An entry of '1' at the *i, j* position represents that the i^{th} platform contains the j^{th} component.

$|I|$ (rows *i*), $|J|$ (columns *j*)

1	0		1	1	1	0	0	0
1	0	….	0	0	0	1	0	0
1	1	….	1	1	0	0	1	1

Figure 5.1 Solution representation for $|I|$ number of platforms

5.3.3 Generation of a Feasible Chromosome Population

Initially a population of solutions is randomly generated; therefore some of the solutions may not be feasible. Also, applying the crossover, and mutation operators create new solutions that may not be feasible.

The feasibility and correction algorithm is slightly different for multiple-platform model as apposed to the single platform model as products that would be assigned to each of the platforms in the platform-set is not known before hand. Therefore for the case of multiple-platform model each of the platforms in a platform-set is checked for feasibility using the *PARs* of the products in the product family as apposed to using the Overall *PAR* for the whole product family. A feasible solution is a solution in which the configuration of each platform follows the part assembly relationship of at least one of the products in the product family.

The feasibility and correction algorithm for the multiple-platform model follows the following steps.

Let P_i (a binary string) represents a configuration of platform '*i*' generated randomly. In the solution representation matrix this string P_i is represented by the i^{th} row, columns 1..., $|J|$.

Let T_j^k (a binary string) represents the j^{th} column in the PAR matrix for product '*k*'. (For an example of PAR matrix, see Table 1).

Step # 1: Set *i*=0, *k*=1

Step # 2 update *i*=*i*+1; if i=$|I|$ go to step 4, else go to Step # 3.

Step # 3: determine strings $Q_j^i = P_i\ AND\ T_j^k$; for all *j*=1in P_i

Step # 4: compare Q_j^i with T_j^k bitwise for all *j*

Step # 5: if for any *j*, Q_j^i not equals T_j^k, go to Step # 6; otherwise platform 'i' is feasible, go to Step # 2.

Step # 6: Update *k*=*k*+1; if *k*=$|K|$ go to Step 7, else go to Step # 3.

Step # 7: Create a new feasible P_i' = [bitwise ***OR*** (T_j^k s)] ***OR*** P_i for all *j*=1 in P_i, go to Step # 2. (where P_i*'* is the corresponding feasible platform configuration for any infeasible P_i).

Step # 8: Stop

5.3.4 Solution Evaluation

The fitness value (f_p) of each solution in the population is calculated to assess the quality of the solution relative to the rest of the solutions in the population. The selections of individuals that are transferred into the next generation are based on their fitness values.

The cost calculations:

Step # 1: For each product *k* produced using platform *i* in the solution calculate the cost of making the platform as:

$$C_{ki} = \left(\sum_{j \in J} \left(CP_j + C_j \right) \cdot x_{ij} + \sum_{j \in J} \left(CA_j + C_j \right) \cdot a_{ijk} + \sum_{j \in J} \left(CR_j - C_j \right) \cdot r_{ijk} \right) D_k + A_i$$

Where, x_{ij}= 1, if component *j* is in platform *i*; a_{ijk} =1, if component *j* is required in product *k* and is not in platform *i*; r_{ijk}= 1, if component *j* is not required in product *k* and $CR_j < C_j$ (it pays to remove the component and use it later).

Step # 2 Construct a square matrix, $M_{|k| \times |k|}$, of elements $C_{ki}s$, Where *M* is the matrix made by concatenating matrices $M_1 = \left[C_{ki} \right]_{|k| \times |I|}$ and $M_2 = \left[\min_{m \in I} \{ C_{km} \} \right]_{|k| \times (|k| - |I|)}$ side-by-side. Figure 5.2 gives an example matrix, *M* of 4 products and 2 platforms. Elements in all rows and the first two columns (M_1) represent the total cost of making product *k* using platform *i*. Two more dummy columns (M_2) are added in which every element in the row is the minimum of elements in the same row of M_1.

Step # 3 Solve the assignment problem represented by matrix $M_{|k|\times|k|}$ for optimal assignments of products to platforms, given the components set of each platform (see Figure 5.2b). Hence for the example in Figure 5.2a, products numbered 1 and 4 would be assigned to platform #1, products numbered 2 and 3 would be assigned to platform #2.

Step # 4 The sum of all $C^*_{ki}s$ gives the cost value of a chromosome.

	Plat 1	Plat2	d1	d2
Prod 1	2	3	2	2
Prod 2	4	3	3	3
Prod 3	1	2	1	1
Prod 4	3	4	3	3

(a)

→

	Plat 1	Plat 2	d1	d2
Prod 1	**C_{11}***	C_{12}	C_{11}	C_{11}
Prod 2	C_{21}	C_{22}	**C_{22}***	C_{22}
Prod 3	C_{31}	**C_{32}***	C_{31}	C_{31}
Prod 4	C_{41}	C_{42}	C_{41}	**C_{41}***

(b)

Figure 5.2 (a) Matrix *M* used for product-platform assignment (b) The final assignment

5.4 Results and Discussions

We use the example presented in Section 3.2 to illustrate and numerically validate the model and the approach presented in this chapter. Several instances of the problem are solved by the exact method (using OPL 3.5) and by the genetic algorithm. Initially, the GA based algorithm was run several times to make the appropriate choice of population size, number of generations, and mutation rate. Based on these runs, a population size of 20, a maximum generation limit of 400, a crossover probability of 0.8, and the mutation probability of 0.1 were used for the remainder of the runs.

Comparative study: The results obtained, from the OPL 3.5 and the Genetic Algorithm approach for the various instances of the problem, are shown in Table 5.2.

The results obtained for the family of drills is consistent with the results obtained for the hypothetical smaller case presented in Section 3.2. The time required to solve the problem using the GA based approach is not presented as for any instance since it was less than 120 seconds. From the results it is obvious that the exact method can require substantial computational time.

As expected, the time required by the exact method increases exponentially with increase in the number of platforms. A comparison of the solution quality using both approaches is presented in Table 5.3 (for the same results presented in Table 5.2). As demonstrated by that table, the heuristic approach reaches a very good solution – within less than 2.6 percent from the optimal results in a reasonable amount of time.

Table 5.2 Results obtained for the various instances of the drill case

Setup cost	Costs (CP_j CA_j CR_j)	Demand	Solution by OPL 3.5			Solution by GA based approach	
			Platforms	Cost ($)	Time (sec.)	Platforms	Cost ($)
$1500	(2 4 3)	[100 100 100 100 100]	[1 6 9 10 11](All products)*	31875	13700	[1 6 8 9 10 11] (All products)	32450
	(2 1.75 3)	[100 100 100 100 100]	[-] **	23875	130	[-]	23875
		[300 50 50 50 50]	[-]	26188	130	[-]	26188
$1000	(2 4 3)	[100 100 100 100 100]	[1 7 9 10 11 22] (P4, P5) [1 4 6 8 9 10 11 16 21] (P1, P2, P3)	30850	36550	[1 3 5 7 9 10 11 14 15](P1, P5) [1 4 6 8 9 10 11] (P4, P2, P3)	31400
		[25 125 125 125 100]	[1 7 9 10 11 22] (P4, P5) [1 4 6 8 9 10 11 16 21] (P1, P2, P3)	29765	37600	[1 2 4 6 8 9 10 11 12 14 17 20 21 22] (P5) [1 4 6 8 9 10 11 16 21] (P2, P3) [1 2 5 6 9 10 12 13 14 16 17 18 19 20 21 22 23] (P1, P4)	30538

*[ ]( ) = [ ] shows platform configuration and ( ) respective assignment of products to the platforms

** [-] Shows the platform does not contain any component or null platform

Table 5.3 Comparing Solutions for the Exact Method and the Heuristic Approach

Setup cost	Costs (CP_j CA_j CR_j)	Demand	Solution by OPL 3.5	Solution by GA based approach	% Difference (GA base approach vs. exact method)
			Cost ($)	Cost ($)	
$1500	(2 4 3)	[100 100 100 100 100]	31875	32450	1.9
	(2 1.75 3)	[100 100 100 100 100]	23875	23875	0
		[300 50 50 50 50]	26188	26188	0
$1000	(2 4 3)	[100 100 100 100 100]	30850	31400	1.86
		[25 125 125 125 100]	29765	30538	2.6

5.4.1 Analysis Using the Genetic Algorithm Results

Figure 5.3 presents the surface plot of the total production cost as a function of setup costs and number of platforms. In this plot the demand of each product (drill) is considered to be 100 units, and CP_j=\$1, CA_j=\$4, and CR_j=\$3 (for all j). The plot shows that the when the setup cost of the platform increase, it is more economical to reduce the number of platforms. When setup cost is reduced, having more platforms result in lower production cost. We can see from the figure that allowing multiple platforms with lower setup costs leads to lower total production costs.

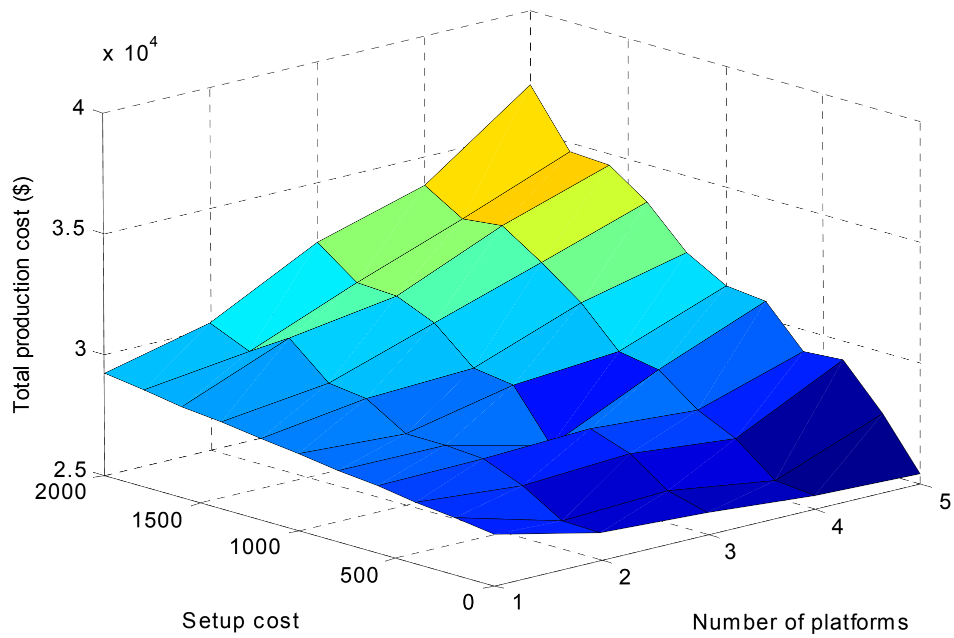


Figure 5.3 Surface plot of Setup cost, number of platforms and total production cost

A convergence plot for the genetic algorithm is shown in Figure 5.4 for different numbers of platforms. The graph shows the solution quality as a function of the number of generations. It is evident that the algorithm converges efficiently.

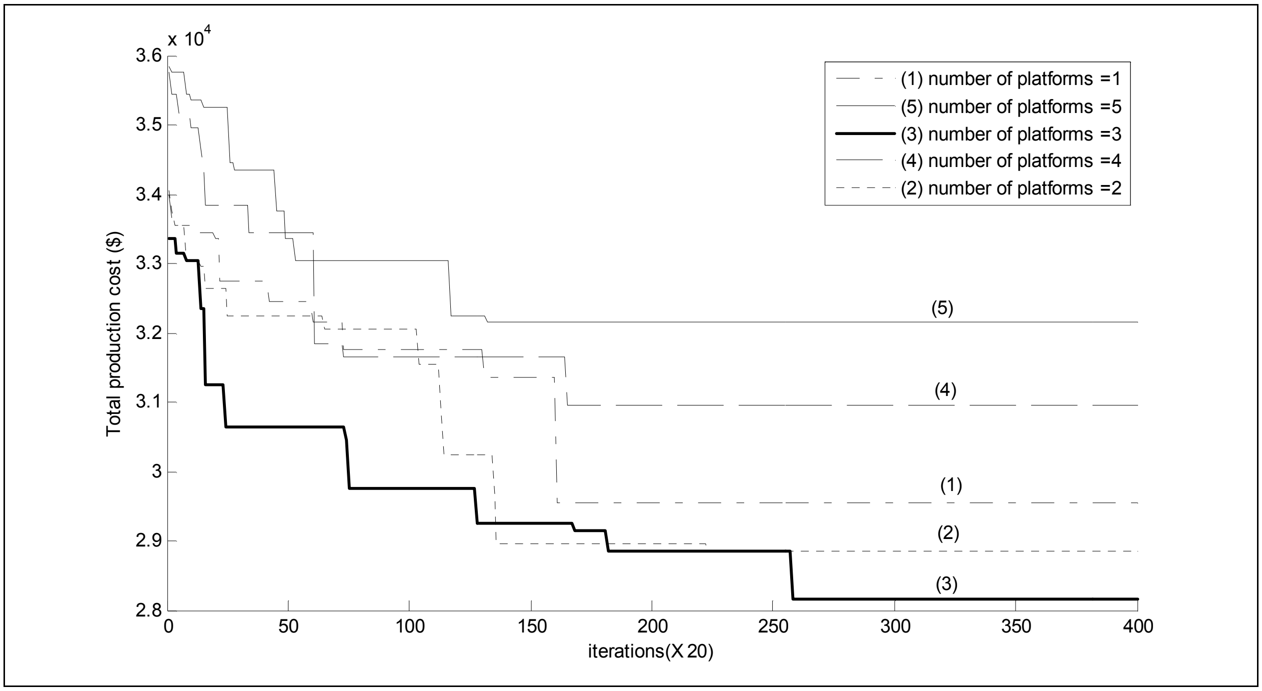


Figure 5.4 Convergence plot for setup cost =$1000

5.5 Conclusions

This chapter introduces the concept of multiple platforms as a solution to the production of a family of products in a cost effective manner. The model establishes that using multiple platforms to produce a family of products, given low setup costs, is cheaper than using a single platform. The chapter presents a description of the problem followed by a mixed integer formulation presented as an optimization problem. Then an evolutionary strategy based on Genetic Algorithm is proposed for the problem. The approach is explained and illustrated with an example of a family of cordless drills. The chapter provides insight into the effects of demand variance and various cost components on the optimal configuration of the platform, and discusses the effectiveness of the heuristic tailored for that application.

CHAPTER 6 - Economic Production of a Product Family under Demand Uncertainty

6.1 Introduction

In this chapter, we propose a platform based approach for the production of a product family under demand uncertainty. Using this approach, every product variant in the family may either be assembled directly from its components, or from any platform whose component set resembles those required by the product. The methodology seeks to minimize the overall production costs of the products, which include the costs of production, and holding cost of unused platform inventory and shortage cost of lost demands of products, while considering the stochastic demand of each product type.

The advantage for using this platform based approach is that this approach enables the economic production of customized products with much shorter final assembly lead times and with decreased risk of losing demand or holding surplus inventory.

The problem is formulated as a two stage stochastic programming model with recourse. First stage decision variables determine the configuration (components set of a platform), and the quantity of the platforms (inventory level) to be produced. Second stage decision variables determine the additional components that would be added to the platform to make a particular product type, the components that would be removed from the platform to make a particular product type, and the quantity of each product type to be produced.

The platform formation problem for the economic production of a product family with stochastic demand of each of the product is modeled as a general optimization problem. The chapter presents three solution approaches to the problem – an exact solution and two heuristic approaches. The results obtained from the exact method are used to validate the model formulation and measure the significance of using stochastic program for modeling the problem. However, only a very small instance of the problem could be solved by exact approach using OPL 3.5. Therefore, two heuristic methods that can provide good solutions to large instances of the problem more quickly are developed. The first heuristic method combines a genetic search process and integer programming to provide a near optimal solution. The heuristic can solve large instance of the problem in a reasonable time. However, with this approach the solution time increases exponentially with increase in number of possible demand scenarios. To deal with large number of demand scenarios a pure probability based genetic search process is proposed, which is very fast even when a large number of demand scenarios is considered with slightly inferior solution quality than that of the first heuristic method. Both of these methods are illustrated with an example of a product family of cordless drills. The example is used to provide insights to the effect of demand parameters and various cost components on the platform based production approach. Also, the heuristics' solution qualities are discussed by comparing the two approaches.

6.2 The Model

In this section, we model and analyze the production of a family of products using platforms that enables cost effective production with short final assembly time in an uncertain demand environment. This model enables the determination of the optimal configuration of product platform (its component set), the optimal inventory level of platform, and the optimal number of each of the product that should be produced in each scenario, while minimizing the overall production costs.

6.2.1 The Problem Statement

A production facility produces N types of products using the platforms (semi-finished form of the products). The facility mass-produces the single type of platforms and keeps the inventory of them.

The manufacturer experiences stochastic demand for each of the products. When the order of a product comes in, some components may be manually added to the platform or some components may be removed from it or both to make the product, and the product is shipped to the customer within the due date. If the actual total demand of all the product types is more than inventory level of the mass produced platforms, there would be some demand losses and shortage cost would be incurred on the other hand, if the actual total demand of the product is less than the inventory level of the platforms, all the demands would be satisfied, but the facility would have to pay holding cost of unused platforms.

The problem can be formulated as a two stage stochastic programming model with recourse. The demand of each product is modeled as set of demand scenarios each with some probability of occurrence.

First stage decision variables would be to:

- Decide the configuration (components set of a platform)
- Decide the quantity of the platforms (inventory level) to be produced

Second stage decision variables would be to:

- Decide the additional components that would be added to the platform to make a particular product type
- Decide the components that would be removed from the platform to make a particular product type
- Decide the quantity of each product type to be produced for each scenario

The objective is to minimize the total production cost that includes the cost of production of platforms, cost of production of products using the platforms, holding cost of unused platforms and stock-out cost of lost demands.

6.2.2 The Model Formulation

A production facility produces *N* types of products using the platforms (semi-finished

The following model (Model 1) of the problem is proposed.

Model 1

$$Min \quad w \times \sum_{j=1}^{M} \left(CP_j + C_j\right) \cdot x_j +$$

$$\sum_{s=1}^{S} p_s \left(\sum_{k=1}^{N} \sum_{j=1}^{M} \left(CA_j + C_j\right) \cdot a_{jk} \cdot y_{ks} + \sum_{k=1}^{N} \sum_{j=1}^{M} \left(CR_j - C_j\right) \cdot r_{jk} \cdot y_{ks} \right) +$$

$$\sum_{s=1}^{S} p_s \sum_{k=1}^{N} q_k \times u_{ks}^{-} +$$

$$\sum_{s=1}^{S} p_s \times h \times v_s^{+} \tag{6.1}$$

S.t.

$$a_{jk} + x_j \le 1 \qquad \forall j \in \{1,2,...,M\}, k \in \{1,2,...,N\} \qquad (6.2)$$

$$a_{jk} + x_j \ge v_{jk} \qquad \forall j \in \{1,2,...,M\}, k \in \{1,2,...,N\} \qquad (6.3)$$

$$v_{jk} \ge a_{jk} \qquad \forall j \in \{1,2,...,M\}, k \in \{1,2,...,N\} \qquad (6.4)$$

$$x_j \ge r_{jk} \qquad \forall j \in \{1,2,...,M\}, k \in \{1,2,...,N\} \qquad (6.5)$$

$$r_{jk} + x_j + v_{jk} \le 2 \qquad \forall j \in \{1,2,...,M\}, k \in \{1,2,...,N\} \qquad (6.6)$$

$$w - v_s^+ = \sum_{k=1}^{N} y_{ks} \qquad \forall s \in \{1,2,...,S\} \qquad (6.7)$$

$$y_{ks} + u_{ks}^- = \xi_{ks} \qquad \forall s \in \{1,2,...,S\}, k \in \{1,2,...,N\} \qquad (6.8)$$

$$1 + x_j \ge f_{jl} + x_l \qquad \forall j,l \in \{1,2,...,M\} \qquad (6.9)$$

$$x_j \in \{0,1\};\ a_{jk} \in \{0,1\};\ r_{jk} \in \{0,1\};\ y_{ks} \ge 0;\ w \ge 0;\ u_{ks}^- \ge 0;\ v_s^+ \ge 0 \qquad (6.10)$$

The objective function (equation 6.1) represents the total production cost that includes cost of making of the platforms, cost of assembling the products using the platforms, total stock-out costs, and total holding cost under all possible scenarios. Constraints 6.2 – 6.4 state that component *j* must be added to the platform to make product *k* if *j* is not in the platform and is required in product *k*. Constraints 6.5 -6.6 state that component *j* may be removed from the platform to make product *k* if that component is in the platform and is not required in product *k*. Constraints 6.7 express that for any scenario *s,* the total number of products produced cannot exceed platform inventory level. Constraints 6.8 limit the total quantity of product *k* produced to the random demand value of product *k* for any scenario *s*. Constraints 6.9 check the assembly feasibility of the platform while deciding the configuration of platform. These constraints states that if component *l* is in the platform and according to part assembly relationship matrix if *j* precedes *l* (f_{jl} =1) then *j* must also be present in the platform. Constraints 6.10 ensure the binary and non-negativity nature of the decision variables.

6.2.3 An Illustrative Example

A production facility produces *N* types of products using the platforms. In this section, a small hypothetical example is used to illustrate the solution of the integer program and to validate the stochastic model by calculating the *stochastic solutions*, *expected value solutions*, and *solutions in case of perfect information*. The model is validated by showing that the *value of stochastic solutions*, VSS, (*expected value solution* - *stochastic solution*) and *expected value of* *perfect information*, EVPI, (*stochastic solution* - *solution in case of perfect information*) are positive for various instances of the example.

Stochastic solutions are determined by solving the stochastic integer program presented in Model 1. Expected value solutions are determined by taking the value of w (number of platforms to be mass produced) equal the sum of the expected demand of all the products and solving the stochastic integer program with this fixed value of w. The solution in case of perfect information is determined by solving the model by taking one scenario at a time with a given demand value of that scenario and the cost value is obtained for that scenario; then weighted sum of the costs for the all the scenarios, where the weight of a scenario equals the probability of the occurrence of that scenario, gives the cost in case of the perfect information.

The example uses a family of three products (*P1, P2*, and *P3*). The binary bills of materials of the products and *PAR* are shown in Table 6.1 and Table 6.2 respectively.

Table 6.1 Binary bill of materials (v_{jk}) for the products

	A	B	C	D
P1	1	1	0	1
P2	1	1	1	0
P3	1	1	0	0

Table 6.2 The *Overall PAR* (f_{jl}) for Product family

	A	B	C	D
A		1	1	
B				1
C				
D				

Data common for all the cases are presented below. The cost of the components is as follows.

Component →	A	B	C	D
Cost ($)→	10	11	12	13

Cost of assembling the platform = $2 per component for all the components.

Cost of adding components to the platform = $4 per component for all the components.

Cost of removing components form the platform = $2 per component.

This small example is solved exactly using OPL 3.5. The reason for taking such a small size problem was that the OPL 3.5 took over 40 hours to solve the problem of this size. This observation motivated us to propose heuristic based approaches to for large real size problems.

Table 6.3 provides the solutions for the various cases of demand scenarios, various shortage costs and holding cost, and various probabilities of occurrence of scenarios. From Table 6.3, following observations are made:

- The positive values of VSS and EVPI (see last two columns of Table 6.3) supports the correctness of the model and it is obvious that there is an advantage of using stochastic model over expected solution approaches.
- When the probability of occurrence of a particular scenario is high the solutions tends to shift towards that scenario (Cases 1, 2, 3, and 5) except for the case of expected value solutions. For a very symmetric case (Case # 4) all the three types of solutions are same, which means for near symmetric cases using expected value solution approach would work well.

Table 6.4 provides a sensitivity analysis on the holding cost and shortage cost using various cases.

From Table 6.4, following observations can be made:

- When the total demands of products are similar in various scenarios then the number of products that should be made in each scenario depends solely on the shortage costs of the products (Case # 1). Also, the shortage cost should be sufficiently high to justify the production of products, as we have not considered the profit of production in our model (See Case # 2).
- When there is high variability in total demand in different scenarios the increase in holding costs encourage lower production for given shortage costs (See Case # 3 and 4).

Table 6.3 Various solutions for the different instances of the example

Case # 1		q_1	Q_2	Q_3	***Expted value sol.***			***Stochastic sol.***			***Perfect information***			VSS	EVPI
h= $ 5		$ 200	$ 100	$ 100	Obj. val. = 9086 w=240			Obj. val.= 8790 w=250			Obj. val. = 8490 w1=250; w2 = 200				
Scenarios	Pr.	D1	D2	D3	Y1	Y2	Y3	Y1	Y2	Y3	Y1	Y2	Y3		
S1	0.8	100	50	100	100	50	90	100	50	100	100	50	100	$296	$300
S2	0.2	50	100	50	50	100	50	50	100	50	50	100	50		
Case # 2		q_1	Q_2	Q_3	Obj. val. = 6606 w=240			Obj. val. = 6330 W=150			Obj. val. = 6014 w1=200;w2= 200				
h= $ 5		$ 200	$ 100	$ 1											
Scenarios	Pr.	D1	D2	D3	Y1	Y2	Y3	Y1	Y2	Y3	Y1	Y2	Y3		
S1	0.8	100	50	100	100	50	90	100	50	0	100	50	100	$276	$316
S2	0.2	50	100	50	50	100	50	50	100	0	50	100	50		
Case # 3		q_1	Q_2	Q_3	Obj. val. = 12840 w=220			Obj. val. = 10365 W=200			Obj. val. = 7595 w1=500; w2=200				
h= $ 50		$ 100	$ 100	$ 100											
Scenarios	Pr.	D1	D2	D3	Y1	Y2	Y3	Y1	Y2	Y3	Y1	Y2	Y3		
S1	0.1	200	200	100	150	70	0	0	100	100	200	200	100	$2475	$2770
S2	0.9	50	100	50	50	100	50	50	100	50	50	100	100		
Case # 4		q_1	Q_2	Q_3	Obj. val. = 8725 w=250			Obj. val. = 8725 W=250			Obj. val. = 8725 w1=250; w2=250				
h=$ 50/80/100		$ 100	$ 100	$ 100											
Scenarios	Pr.	D1	D2	D3	Y1	Y2	Y3	Y1	Y2	Y3	Y1	Y2	Y3		
S1	0.5	100	50	100	100	50	100	100	50	100	100	50	100	$0	$0
S2	0.5	50	100	100	50	100	100	50	100	100	50	100	100		
Case # 5		q_1	Q_2	Q_3	Obj. val. = 20368 w=470			Obj. val. = 18835 W=500			Obj. val. = 16265 w1=500; w2 =200				
h= $ 5		$ 100	$ 100	$ 100											
Scenarios	Pr.	D1	D2	D3	Y1	Y2	Y3	Y1	Y2	Y3	Y1	Y2	Y3		
S1	0.9	200	200	100	200	170	100	200	200	100	200	200	100	$1533	$2570
S2	0.1	50	100	50	50	100	50	50	100	50	50	100	50		

Table 6.4 Sensitivity analysis on holding costs and shortage costs

Case # 1		***q₁***	***q₂***	***q₃***	***Stochastic sol.***		
h= \$50/ \$80/ \$100		\$ 100	\$ 100	\$ 100	Obj. val. = \$8790 w=250		
Scenarios	Pr.	D1	D2	D3	Y2	Y2	Y2
S1	0.8	100	50	100	100	50	100
S2	0.2	50	100	50	50	100	50
Case # 2		***q₁***	***q₂***	***q₃***	Obj. val. = 5000 W=0		
h= \$50/ \$80/ \$100		\$ 20	\$ 20	\$ 20			
Scenarios	Pr.	D1	D2	D3	Y2	Y2	Y2
S1	0.5	100	50	100	0	0	0
S2	0.5	50	100	50	0	0	0
Case # 3		***q₁***	***q₂***	***q₃***	Obj. val. = 18825 w=320		
h= \$ 50		\$ 102	\$ 101	\$ 100			
Scenarios	Pr.	D1	D2	D3	Y2	Y2	Y2
S1	0.5	200	100	100	200	100	20
S2	0.5	100	50	50	100	50	50
Case # 4		***q₁***	***q₂***	***q₃***	Obj. val. = 27830 w=200		
h=\$100		\$ 102	\$ 101	\$ 100			
Scenarios	Pr.	D1	D2	D3	Y2	Y2	Y2
S1	0.5	200	100	100	200	0	0
S2	0.5	100	50	50	100	50	50

6.3 Solution Approaches

We propose genetic evolutionary based solution methodologies for the problem as a genetic algorithm based heuristics are vastly used tool for optimization problems that have large search spaces and with non-linear objective functions. The problem presented in this chapter falls in that category. Usually such heuristic approaches are required to solve stochastic models (Spall, 2003).

For this model, we proposed two types of heuristic approaches. The first heuristic method, Genetic Algorithm with Integer Programming (GAIP), combines the genetic search process and integer programming to provide a near optimal solution. The heuristic can solve a large instance of the problem in a reasonable time; however the solution time increases exponentially with increase in number of possible demand scenarios. The second heuristic method, multiple-population Genetic Algorithm, is a pure probability based heuristic search process (called PHA in rest of the thesis) that starts with multiple populations, and can solve the instances of the problem with large number of demand scenarios.

The detailed explanations of the heuristic approaches are provided in sections 6.4 and 6.5. However, the chromosome encoding and feasibility check process is the same for both approaches. Also, the Solution Encoding approach and the Initial Feasible Population Generation (feasibility and correction of platform configuration) algorithm used for the heuristic approaches are identical to what used in solution methodology in Chapter 4, Section 4.3.1. Both the heuristic solution approaches are presented in the following sections.

6.4 Genetic Algorithm with Integer Programming (GAIP)

In this method a genetic search algorithm, presented in Figure 6.3, is used to explore the search space and an integer program (Model 2), presented in Section 6.4.1, is solved for each chromosome to calculate its fitness value.

The solution methodology follows the strategy shown in the flowchart in Figure 6.1. A chromosome string, x_j , is generated probabilistically (randomly or using genetic operators) and value of x_j is fed to the Model 1 and the model reduces to a linear integer program (Model 2). Solving this integer program provides the total cost of production (objective function value of Model 2) and values of other decision variables such as the number of platforms to be mass-produced, the components that should be added or removed to produce a particular product, the number of each product produced using the mass-produced platforms. Now the genetic operators are applied to alter the chromosome information (x_j) and then the new value of x_j is fed back to Model 1 to reduce it to Model 2 and Model 2 is solved to get new objective function value and the solutions. This process is repeated until maximum number of iterations is reached.

Clearly, this approach combines the genetic search process and solving an integer program. The genetic search process and solution process of integer program is explained in detail in the following section.

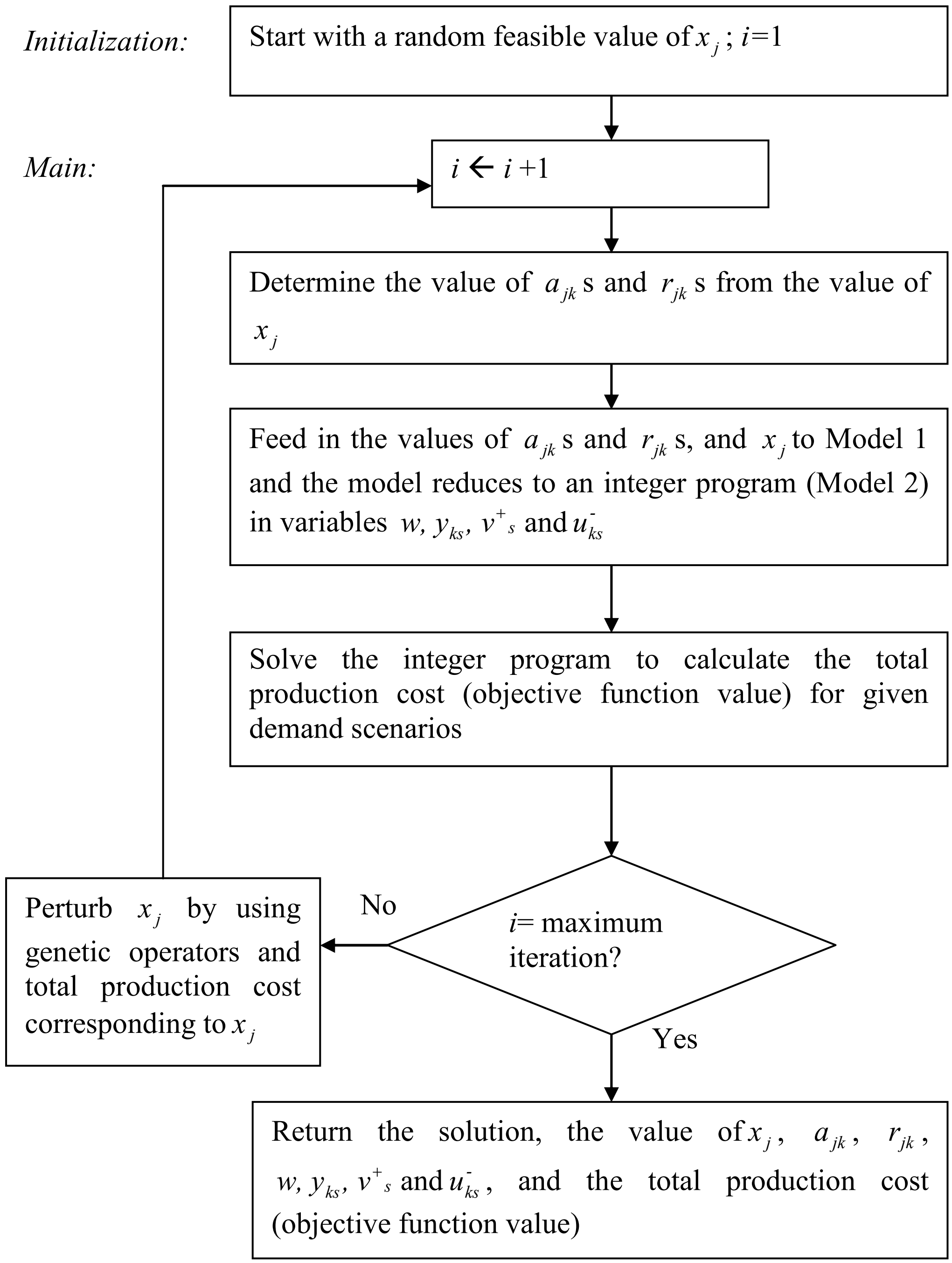


Figure 6.1 The flow chart showing the solution strategy

6.4.1Genetic Search Process

The search process starts with the encoding and the initial feasible solution generation, which is explained in Section 4.1.1 and Section 4.1.2. The overall process is presented by a flow chart shown in Figure 6.3. The details of the various steps presented in the flow chart are as follows.

Perturbing the values of x_j s (Crossover and Mutation)

The crossover used for the exploration of search space. If the population size is 2*L, L* random pairs of chromosomes are selected for crossover. The crossover operator is applied on each of these pairs with crossover probability (here 0.8-0.9). The pairs that undergo crossover generate that many pairs of children. The crossover operator used here is single point crossover. In a single point crossover a *crossover point* (See Figure 6.2) is randomly selected on the pair of chromosomes undergoing crossover and the bits on the chromosomes are exchanged about that crossover point. Figure 6.2 explains the single point crossover operation.

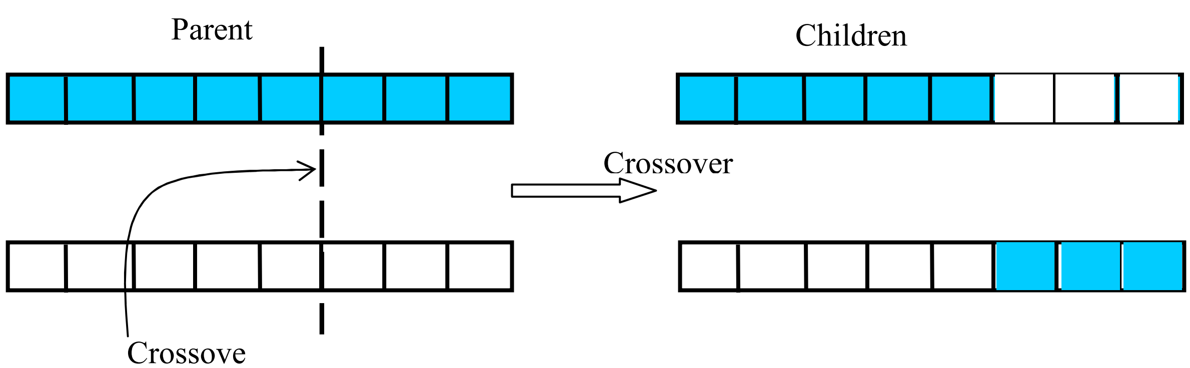


Figure 6.2 Crossover

The mutation process used here is identical to what presented in Section 4.3.2. The mutation is applied on a string (chromosome) with a very low probability.

Next the children population and the mutant population obtained after the application of crossover and mutation operators are checked for the feasibility. If any chromosome (string) is not feasible, we modify the string to become feasible using *Feasibility Check and Correction Algorithm* (the algorithm is presented in Section 4.3.1).

Building each Generation

The algorithm starts with the random generation of *L* solutions. The feasibility of each solution is checked and if not feasible they are made feasible. These solutions are evaluated and kept in an archive. Then these solutions and a copy of each of them are taken for crossover and mutation. After this step we get altered solutions (children and mutants). These solutions are checked and corrected for feasibility. These mutated feasible solutions are evaluated and the best *L* of the (Archive solutions + children and mutants solutions) updates the archive. The process is repeated until the pre-specified maximum number of iterations is reached.

The population size is doubled during the crossover and mutation phase to make the search more exploratory.

Evaluation and Calculation of Fitness

Each suggested platform has a cost function that shows the cost of converting the platform to each of the products, given relevant data (various associated costs, demand in each scenario, the probability values for each scenarios, etc.).

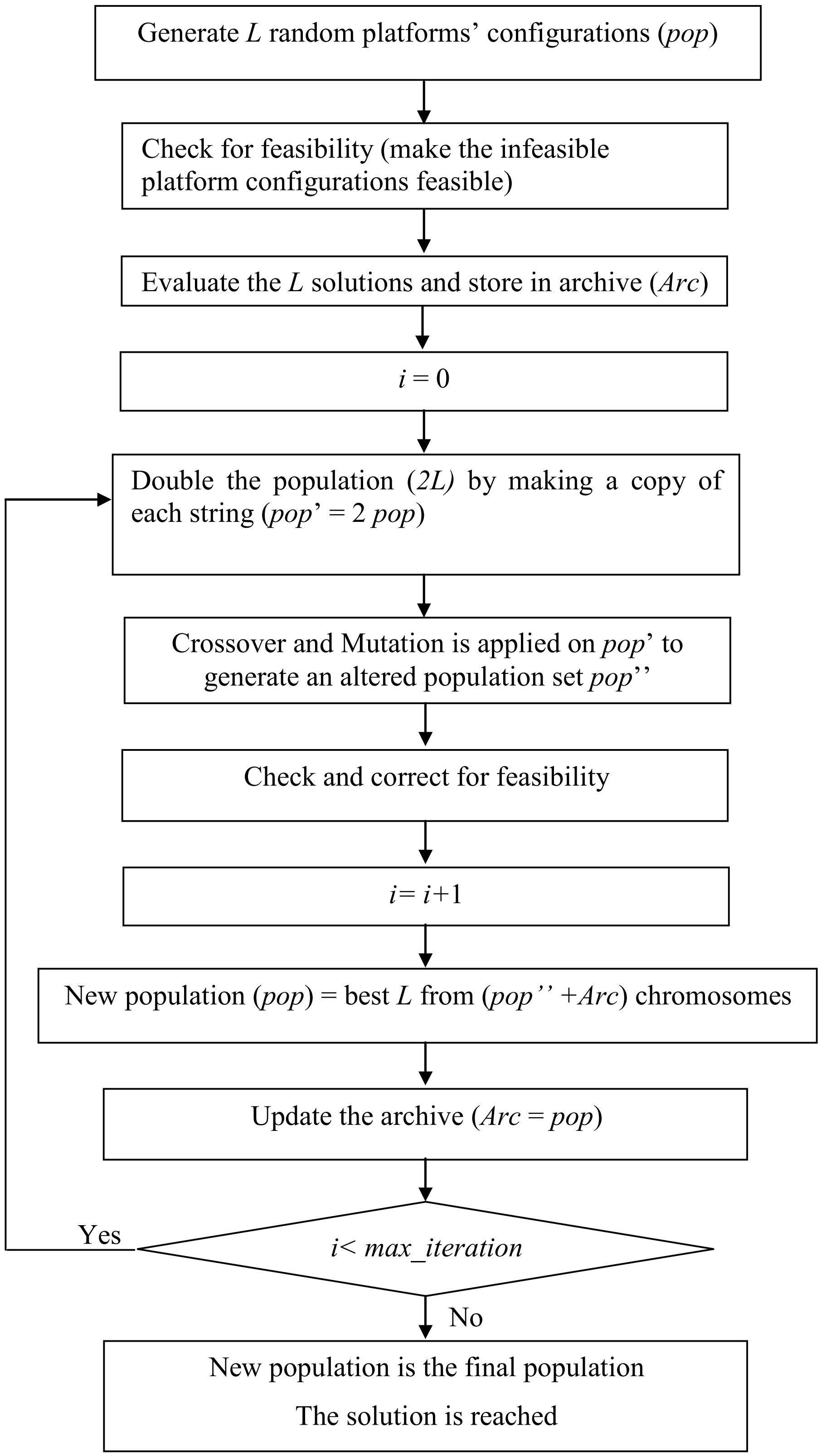


Figure 6.3 The flow chart of the genetic search algorithm

The cost calculations follow the following steps:

Step #1 From a suggested value of x_j, determine the values of a_{jk} and r_{jk}. How this is done is shown in the following pseudo-code below.

For each component j in the platform ***Do***

For each product i ***Do***

If *j is NOT in product i* ***Then***

If *$CR_j > C_j$* ***Then***

$r_{jk} = 0$

Else

$r_{jk} = 1$

End *i*

End *j*

For each component j Not in platform ***Do***

For each product i ***Do***

If *j is in the product* ***Then***

$a_{jk} = 1$

Else

$a_{jk} = 0$

EndIf

End *i*

End *j*

Step #2 Reduce Model 1 to an integer programming model (Model 2) by putting in the values of x_j, a_{jk} and r_{jk}. The reduced model 2 is shown below.

Model 2 (Underlined variables are known)

$$Min \quad Z = \quad w \times \sum_{j=1}^{M} \left(CP_j + C_j\right) \cdot \underline{x_j} +$$
$$\sum_{s=1}^{S} p_s \left(\sum_{k=1}^{N} \sum_{j=1}^{M} \left(CA_j + C_j\right) \cdot \underline{a_{jk}} \cdot y_{ks} + \sum_{k=1}^{N} \sum_{j=1}^{M} \left(CR_j - C_j\right) \cdot \underline{r_{jk}} \cdot y_{ks} \right) +$$
$$\sum_{s=1}^{S} p_s \sum_{k=1}^{N} q_k \times u_{ks}^{-} +$$
$$\sum_{s=1}^{S} p_s \times h \times v_s^{+} \tag{6.11}$$

$s.t.$

$$w - v_s^{+} = \sum_{k=1}^{N} y_{ks} \qquad \forall\ s \in \{1,2,\ldots,S\} \tag{6.12}$$
$$y_{ks} + u_{ks}^{-} = \xi_{ks} \qquad \forall\ s \in \{1,2,\ldots,S\}, k \in \{1,2,\ldots,N\} \tag{6.13}$$
$$y_{ks} \geq 0;\ w \geq 0\ ;\ v_s^{+} \geq 0;\ u_{ks}^{-} \geq 0 \tag{6.14}$$

This integer program is solved to provide the solution and the value of the objective function, which is the total production cost (Z^*). How this integer program is solved is explained in Section 6.4.2.

Step # 3 The fitness of the chromosome is given as $fitness = 1/Z^*$.

6.4.2 Solving the Integer Program (Model 2)

Model 2 is an integer program. In general, solving an integer program is difficult and solutions are obtained in substantial amount of time. However, it can be proved that the constraints matrix of Model 2 is a Totally Unimodular Matrix (TUM) and the right hand side of the all the constraints (b) are integers. Therefore, Model 2 can be solved as a linear program and always integral solutions will be obtained.

Proof for Total Unimodularity of the coefficient matrix of linear constraints of the Model 2:
The constraints of Model 2 can be rewritten as (in $AX = b$ form),

$$w - \sum_{k=1}^{N} y_{ks} + 0 \times u_{ks}^{-} - v_s^{+} = 0 \qquad \forall\, s \in \{1,2,\ldots,S\}$$

$$0 \times w + y_{ks} + u_{ks}^{-} + 0 \times v_s^{+} = \xi_{ks} \qquad \forall\, s \in \{1,2,\ldots,S\}, k \in \{1,2,\ldots,N\}$$

The A matrix will have total number of rows, $m = S + S*N$ and number of columns, $n = 1+S*N + S*N + S$ (n= total number of variables the model).

The matrix A can be presented as $A_{m\times n} = \left(F_{(S+S\cdot N)\times 1}, G_{(S+S\cdot N)\times(S\cdot N)}, H_{(S+S\cdot N)\times(S\cdot N)}, L_{(S+S\cdot N)\times S}\right)$

An example A matrix for number of scenarios (S) = 3 and number of products (N) = 5 is presented below.

	F	**G**										**H**										**L**	
Q₁	1	-1	-1	-1	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0
Q₂	1	0	0	0	0	0	-1	-1	-1	-1	-1	0	0	0	0	0	0	0	0	0	0	0	-1
	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Q₁	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Q₂	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0

F is a matrix with elements, $f_{ij} = \begin{cases} 1 & \text{for } i=1,2,\ldots,S \text{ and } j=1 \\ 0 & \text{for } i=S+1,\ldots,S+S\cdot N \text{ and } j=1 \end{cases}$

(For all $i = 1,\ldots,S + S\cdot N;\ j = 1$)

G is a matrix with elements, $g_{ij} = \begin{cases} -1 & \text{for } i=1,..,S \text{ and } j=1+(i-1)\cdot N+1,\ldots,1+i\cdot N \\ 0 & \text{for } i=1,\ldots,S \text{ and } j\neq 1+(i-1)\cdot N+1,\ldots,1+i\cdot N \\ 1 & \text{for } i=S+1,\ldots,S+S\cdot N \text{ and } j=1+i-S \\ 0 & \text{for } i=S+1,\ldots,S+S\cdot N \text{ and } j\neq 1+i-S \end{cases}$

(For all $i = 1,\ldots,S + S\cdot N;\ j = 1+1,\ldots,1+S\cdot N$)

H is a matrix with elements, $h_{ij} = \begin{cases} 0 \text{ for } i=1,\ldots,S \text{ and } j=1+S.N+1,\ldots 1+S.N+S.N \\ 1 \text{ for } i=S+1,\ldots,S+S.N \text{ and } j=1+S.N+i-S \\ 0 \text{ for } i=S+1,\ldots,S+S.N \text{ and } j\neq 1+S.N+i-S \end{cases}$

L is a matrix with elements, $l_{ij} = \begin{cases} -1 \text{ for } i=1,\ldots,S \text{ and } j=1+S.N+S.N+i \\ 0 \text{ for } i\neq 1,\ldots,S \text{ and } j=1+S.N+S.N+i \\ 0 \text{ for } i=S+1,\ldots,S+S.N \text{ and } j=1+S.N+S.N+1,\ldots 1+S.N+S.N+S \end{cases}$

From the definition of h_{ij} and l_{ij}, *H* and *L* are portions of the identity matrices (*I*). A common result states that B is a TUM if and only if $A = (B, I)$ is also a TUM. Thus it is suffice to show that $B= (F, G)$ is a TUM. We will use a common theorem that states that $T_{m\times n}$ is a TUM if for every $Q \subseteq \{1,..,m\}$, there exist a partition Q_1, Q_2 of Q such that $\left|\sum_{i\in Q_1} e_{ij} - \sum_{i\in Q_2} e_{ij}\right| \leq 1$ for $j = 1,\ldots,n$ to prove that *B* is a TUM.

For the matrix *B*, we choose set

$$Q_1 = \{1,3,5\ldots,(2\cdot i-1),\ldots\}\cup$$
$$\{S+1,\ldots,S+N\}\cup\{S+2\cdot N+1,\ldots,S+3\cdot N\}\ldots\cup\{S+(2\cdot i-2)\cdot N+1,\ldots,S+(2\cdot i-1)\cdot N\}\ldots$$
$$\text{for } \forall\, i = 1,\ldots, S/2 \text{ (if S = an even number) or } S+1/2 \text{ (if S = an odd number)}$$

$$Q_2 = \{2,4,6,\ldots,2\cdot i,\ldots\}\cup$$
$$\{S+N+1,\ldots,S+2\cdot N\}\cup\{S+3\cdot N+1,\ldots,S+4\cdot N\}\ldots\cup\{S+(2\cdot i-1)\cdot N+1,\ldots,S+(2\cdot i)\cdot N\}\cup\ldots$$
$$\text{for } \forall\, i = 1,\ldots, S/2 \text{ (if S = an even number) or } S-1/2 \text{ (if S = an odd number)}$$

(Refer the example matrix *A* presented above to see the partition of rows in two sets Q_1 and Q_2)

Let e_{ij} is an element of *B*, then

$$\left(\sum_{i\in Q_1} e_{ij} - \sum_{i\in Q_2} e_{ij}\right) = \begin{cases} \left(\sum_{i\in Q_1} f_{ij} - \sum_{i\in Q_2} f_{ij}\right) \text{ for } j=1 \\ \left(\sum_{i\in Q_1} g_{ij} - \sum_{i\in Q_2} g_{ij}\right) \text{ for } j=1+1,\ldots,1+S.N \end{cases}$$

From the definition of f_{ij},

$$\sum_{i\in Q_1} f_{ij} = \begin{cases} S/2 \text{ if } S = \text{an even number} \\ S+1/2 \text{ if } S = \text{an odd number} \end{cases}$$

$$\sum_{i \in Q_2} f_{ij} = \begin{cases} S/2 & \text{if } S = \text{an even number} \\ S-1/2 & \text{if } S = \text{an odd number} \end{cases}$$

Therefore, $$\left(\sum_{i \in Q_1} f_{ij} - \sum_{i \in Q_2} f_{ij} \right) = \begin{cases} 0 & \text{if } S = \text{an even number} \\ 1 & \text{if } S = \text{an odd number} \end{cases}$$

From the definition of g_{ij},

$$\sum_{i \in Q_1} g_{ij} = \begin{cases} -1 + \left(S/2 \right) & \text{if } S = \text{an even number} \\ -1 + \left(S+1/2 \right) & \text{if } S = \text{an odd number} \end{cases}$$

$$\sum_{i \in Q_2} g_{ij} = \begin{cases} -1 + \left(S/2 \right) & \text{if } S = \text{an even number} \\ -1 + \left(S-1/2 \right) & \text{if } S = \text{an odd number} \end{cases}$$

Therefore, $$\left(\sum_{i \in Q_1} g_{ij} - \sum_{i \in Q_2} g_{ij} \right) = \begin{cases} 0 & \text{if } S = \text{an even number} \\ 1 & \text{if } S = \text{an odd number} \end{cases}$$

Now, $$\left(\sum_{i \in Q_1} e_{ij} - \sum_{i \in Q_2} e_{ij} \right) = \begin{cases} 0 \text{ or } 1 & \text{for } e_{ij} \in F \\ 0 \text{ or } 1 & \text{for } e_{ij} \in G \end{cases}$$

Or, $$\left| \sum_{i \in Q_1} e_{ij} - \sum_{i \in Q_2} e_{ij} \right| \le 1$$

Which means B is a TUM and hence A is a TUM. □

The Model 2 is solved as a linear program using Simplex method which is very fast.

6.5 Pure Probability based Heuristic Approach (PHA)

The time taken by the GAIP approach, to solve the problem, increases exponentially with the number of demand scenarios. To alleviate this problem we propose one more heuristic which is a pure probability based genetic search process.

This is a multiple population genetic search heuristic. The heuristic starts with proposing a population of chromosomes corresponding to each demand scenario. The underlying premises behind developing this heuristic are as follows.

- Good chromosomes from each population are selected based on their performance (= the total production cost value by using that platform to produce products for the demands in that scenario) with respect to that corresponding demand scenario. The number of good chromosomes selected from a scenario solution is proportional to the value of the probability of occurrence of that scenario to improve the chances of getting more "traits" of those chromosomes, which corresponds to a scenario with high probably of occurrence, in the children population.
- These selected "good" chromosomes are mixed together and they undergo crossover and mutation in anticipation that children chromosomes would demonstrate overall good performance for all the scenarios together. (During the course of the search process the chromosomes are evaluated for their performance in each scenario and overall performance in the entire scenarios).
- The selection process directs the search process. The heuristic uses three levels of selection process. First, the chromosomes are selected on the basis of their performance with respect to a particular scenario and second, among these selected chromosomes and their children chromosomes some chromosomes are selected further on the basis of their overall performance in the entire demand scenarios. Finally, only those chromosomes are selected for next iteration that has good overall performance in entire scenarios and good performance in a particular scenario.

The algorithm for the approach is presented next, before some definitions are provided that are used to explain the algorithm.

Definitions:

1. *The objective value with respect to scenario 's' (Obj_s):* The total cost of production calculated considering that only scenario '*s*' would be realized. For a given configuration of platform (x_j), the calculation of Obj_s follows the following steps:

Step # 1: Using x_j, determine the values of a_{jk} and r_{jk} as presented in Section 6.4.1.

Step # 2: Put in these values in the following equation to get the value of Obj_s.

$$Obj_s = \sum_{k=1}^{N} \sum_{j=1}^{M} \left(C_p + C_j\right) \cdot x_j \cdot D_k + \sum_{k=1}^{N} \sum_{j=1}^{M} \left(\left(C_a + C_j\right) \cdot a_{jk} + \left(C_r - C_j\right) \cdot r_{jk}\right) \cdot D_k$$

D_k is the demand of product k in scenario s.

2. *The expected object value (Obj_E):* The total cost of platform based production calculated considering the expected demand of the each of the product. This cost calculation follows the same steps presented for the calculation of Obj_s except that in place of D_k, expected value of D_k, $E(D_k)$, is used. $E(D_k) = \sum_{s=1}^{S} (p_s \times D_k)$.

The pure probability based genetic search approach follows the following steps:

Step # 1: Corresponding to each scenario a random population (population size = *psize*) of chromosomes is generated. Let pop_s be the population of chromosomes corresponding to scenario *s*. And, Obj_s is calculated for all chromosomes in *Ps*.

Step # 2: Then take best ($[p_s \times pop_s]$+1), [x] is the greatest integer value of x, chromosomes from each pop_s and mix them to make a bigger population called *pop'*.

Step # 3: Double the size of *pop'* by making a copy of it. Perform crossover with high crossover probability and mutation with low probability on doubled *pop'* to generate *children* and *mutants*. The crossover and mutation process is same as explained in Section 6.4.1. This step is performed to explore search-space to look for potential solutions. The *children*, *mutants* and the *pop'* is combined to get a bigger population, *pop''*.

Step # 4: For each chromosome in *pop''* all Obj_s are determined and Obj_E is determined.

Step # 5: Sort the *pop''* by increasing value of Obj_E and take the top *L* (*L*= size of *pop'*) chromosomes and rest of the chromosomes are discarded. The top *L* chromosomes are called *pop'''*.

Step # 6: Take the best chromosome(s) (chromosome with minimum value of Obj_E) from *pop'''*. Use the x_j value of this chromosome to calculate a_{jk} and r_{jk}, and feed these values in Model 1 to get Model 2. Solve Model 2 to get the object value (total production cost) and the solutions corresponding to this chromosome. This is the solution for first iteration.

Step # 7: Sort *pop'''* in increasing order of Obj_s and take the top *psize* chromosomes to get pop_s for the next iteration. Do this for all *s*.

Step # 8: Increment the number of iteration. If number iteration is greater than maximum number of iteration go to Step # 9, otherwise go to Step # 2.

Step # 9: Terminate the algorithm, report the objective value and solutions of the last iteration.

The crossover and mutation processes used in this approach are same as that of the GAIP. These two approaches are used to solve the various instances of the case study presented in next section. Also, some comparative studies are performed to expose the solution qualities of these approaches and their suitability for the different instances of the problem.

6.6 Results and Discussions

Initially, the genetic search heuristics proposed in last section were run several times to make the appropriate choice of population size, number of generations, and mutation rate. The appropriate values of population sizes and maximum number of generations depend upon the various instances of the problem taken. For all the instances and for both the approaches the crossover probability is kept high (0.8-0.9) and the mutation probability is kept low (0.05-0.1).

6.6.1 Results and Analysis using GAIP

The search process starts with the encoding and the initial feasible solution generation,

Table 6.5 provides the comparison of solutions obtained from GAIP with the exact solutions for the small example presented in Section 6.2.3.

Table 6.5 The comparison of solutions obtained from GAIP with the exact solutions

Case # 1		q_1	q_2	q_3	***Stochastic (optimal)***			***Stochastic (GAIP)***		
h= $ 5		$ 200	$ 100	$ 100	Obj. val. = $8790 w=250			Obj. val. = $8790 w=250		
Scenarios	Pr.	D1	D2	D3	Y1	Y2	Y3	Y1	Y2	Y3
S1	0.8	100	50	100	100	50	100	100	50	100
S2	0.2	50	100	50	50	100	50	50	100	50
Case # 2		q_1	q_2	q_3	Obj. val. = 6330 w=150			Obj. val. = 6330 w=150		
h= $ 5		$ 200	$ 100	$ 1						
Scenarios	Pr.	D1	D2	D3	Y1	Y2	Y3	Y1	Y2	Y3
S1	0.8	100	50	100	100	50	0	100	50	0
S2	0.2	50	100	50	50	100	0	50	100	0
Case # 3		q_1	q_2	q_3	Obj. val. = 10365 w=200			Obj. val. = 10365 w=200		
h= $ 50		$ 100	$ 100	$ 100						
Scenarios	Pr.	D1	D2	D3	Y1	Y2	Y3	Y1	Y2	Y3
S1	**0.1**	200	200	100	0	100	100	0	100	100
S2	**0.9**	50	100	50	50	100	50	50	100	50
Case # 4		q_1	q_2	q_3	Obj. val. = 8725 w=250			Obj. val. = 8725 w=250		
h=$ 50/80/100		**$ 100**	**$ 100**	**$ 100**						
Scenarios	Pr.	D1	D2	D3	Y1	Y2	Y3	Y1	Y2	Y3
S1	**0.5**	100	50	100	100	50	100	100	50	100
S2	**0.5**	50	100	100	50	100	100	50	100	100
Case # 5		q_1	q_2	q_3	Obj. val. = 18835 w=500			Obj. val. = 18835 w=500		
h= $ 5		**$ 100**	**$ 100**	**$ 100**						
Scenarios	Pr.	D1	D2	D3	Y1	Y2	Y3	Y1	Y2	Y3
S1	**0.9**	200	200	100	200	200	100	200	200	100
S2	**0.1**	50	100	50	50	100	50	50	100	50

From Table 6.5, it is obvious that the GAIP provides the optimal solutions for the small example. The rest of the results and analysis provided in this section are on the bigger example. Table 6.6 provides the data for an instance of the bigger example. For this example, CP_j=\$2, CA_j=\$4, and CR_j=\$2 (for all j). Table 6.7 presents the solution obtained using the GAIP for the example.

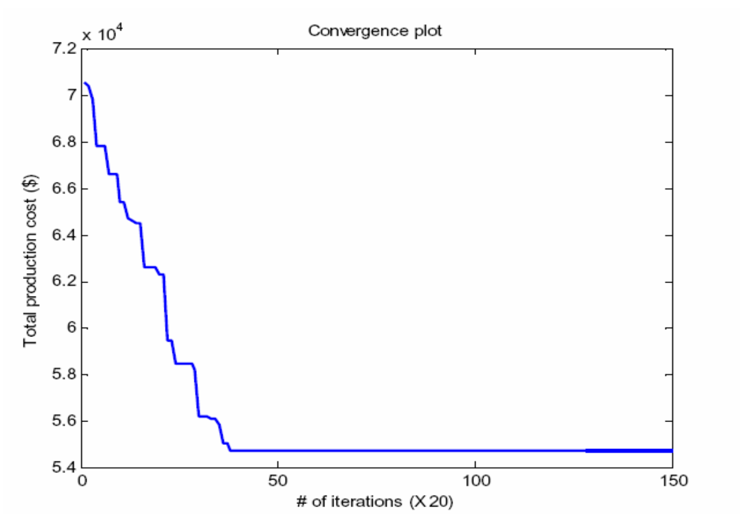
Table 6.6 Data for an example problem

Scenarios	Pr.	Demand				
		P1	P2	P3	P4	P5
S1	0.2	100	100	100	100	100
S2	0.8	200	200	200	200	200
Holding cost (Platforms)		Shortage cost				
		P1	P2	P3	P4	P5
$10		100	100	100	100	100

Table 6.7 Solution obtained for the example using GAIP

Objective value = $ 54710 Number of platforms made = 600											
Scenarios	Number of products made					Shortages of products					Leftover platforms
	P1	P2	P3	P4	P5	P1	P2	P3	P4	P5	
S1	100	100	100	100	100	0	0	0	0	0	100
S2	200	200	0	0	200	0	0	200	200	0	0

Figure 6.4 shows the convergence plot of GAIP for the above case.



. **Figure 6.4 Convergence plot of GAIP**

The graph shows the solution quality as a function of the number of generations. It is evident that the algorithm converges efficiently.

The heuristic is run 20 times, and the mean and 'mean + standard deviation' of best solutions for all the runs at each iteration is plotted *vs.* number of iterations. The plot is shown in Figure 6.5. From Figure 6.5 it is clear that the standard deviation kept decreasing with the number of iterations and finally it becomes negligible which means that in almost all the runs the heuristic hit the same solution at the end which in turn supports the global convergence and repeatability of the heuristic.

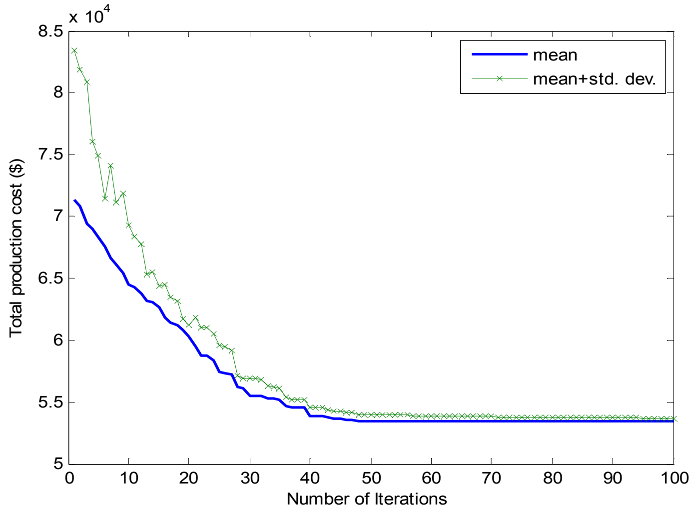


Figure 6.5 The plot of mean and 'mean + standard deviation' of the iteration best of all the runs at each iteration vs. Iterations (using GAIP approach)

6.6.2 The solution quality of PHA and comparison with GAIP approach

Figure 6.6 shows a convergence plot of PHA for any instance of the example. This approach requires more number of iterations than the GAIP, but it converges efficiently too.

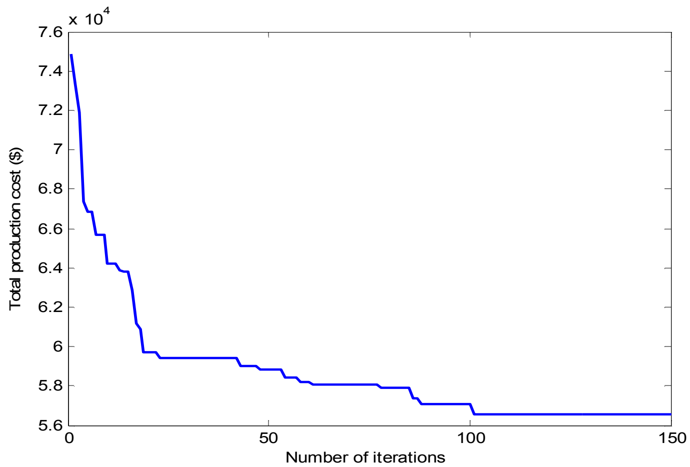


Figure 6.6 A convergence plot for the PHA

The graph shown in Figure 6.5 for the GAIP approach is also plotted for PHA approach and is presented in Figure 6.7. Figure 6.7 supports the global convergence and repeatability of the PHA approach.

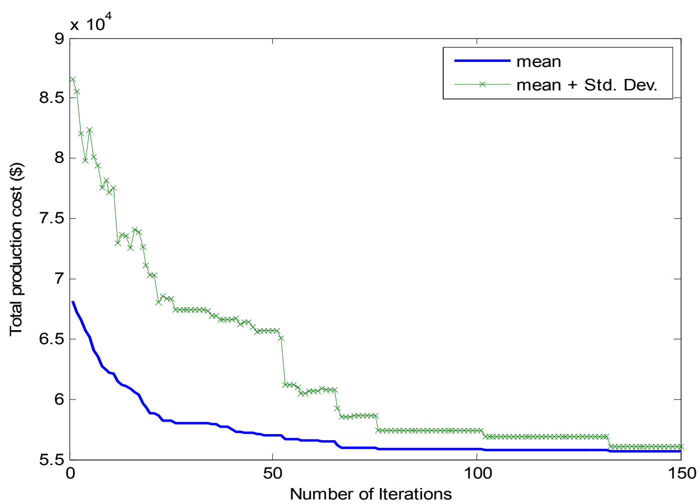


Figure 6.7 The plot of mean and 'mean + standard deviation' of the iteration best of all the runs at each iteration vs. Iterations (using PHA approach)

Table 6.8 shows the comparison of results obtained from both the heuristics. The results obtained from both the approaches prove that GAIP performs slightly better than PHA. Some more comparisons between these two approaches are provided in following section.

Table 6.8 Comparison of results obtained from both the heuristics

Case # 1		q_1	q_2	q_3	q_4	q_5	**GAIP**					**PHA**				
		$ 200	$ 100	$ 200	$ 100	$ 100	W=500; Cost value=26895					W=500; Cost value=27985				
h= $ 5																
Scenarios	Pr.	D1	D2	D3	D3	D5	Y1	Y2	Y3	Y4	Y5	Y1	Y2	Y3	Y4	Y5
S1	0.8	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
S2	0.2	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
Case # 2		q_1	q_2	q_3	q_4	q_5	W=250; Cost value = 19720					W=250; Cost value = 20220				
		$ 200	$ 100	$ 200	$ 100	$ 100										
h= $ 5																
Scenarios	Pr.	D1	D2	D3	D3	D5	Y1	Y2	Y3	Y4	Y5	Y1	Y2	Y3	Y4	Y5
S1	0.8	50	50	50	50	50	100	100	100	100	100	100	100	100	100	100
S2	0.2	100	100	100	100	100	100	0	100	50	0	100	0	100	50	0
Case # 3		q_1	q_2	q_3	q_4	q_5	W=250; Cost value= 21183					W=250; Cost value= 21758				
		$ 100	$ 100	$ 100	$ 100	$ 100										
h= $ 50																
Scenarios	Pr.	D1	D2	D3	D3	D5	Y1	Y2	Y3	Y4	Y5	Y1	Y2	Y3	Y4	Y5
S1	**0.1**	200	200	200	200	200	0	0	0	50	200	0	0	0	50	200
S2	**0.9**	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
Case # 4		q_1	q_2	q_3	q_4	q_5	W=250; Cost value=50525					W=250; Cost value=52025				
		$ 100	**$ 100**	**$ 100**	**$ 100**	**$ 100**										
h=$ 50/80/100																
Scenarios	Pr.	D1	D2	D3	D3	D5	Y1	Y2	Y3	Y4	Y5	Y1	Y2	Y3	Y4	Y5
S1	**0.5**	200	200	200	200	200	0	200	50	0	0	0	0	0	50	200
S2	**0.5**	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
Case # 5		q_1	q_2	q_3	q_4	q_5	W=1000; Cost value=57518					W=1000; Cost value=57368				
		$ 100	**$ 100**	**$ 100**	**$ 100**	**$ 100**										
h= $ 10																
Scenarios	Pr.	D1	D2	D3	D3	D5	Y1	Y2	Y3	Y4	Y5	Y1	Y2	Y3	Y4	Y5
S1	**0.9**	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200
S2	**0.1**	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50

6.6.3 Results when demand of each product is given in terms probability distribution

So far we considered the cases of scenario based demand realization. In this section we would consider that instead of demand scenarios and their probability of occurrence, the demand distribution of each of product is specified. For the cases here we assume that the demand of each of the product follows a normal distribution with some mean and standard deviation.

As the mathematical formulation of the model considers the scenario based demand realization, we would convert the probability distribution based demand information into scenario based demand realization. Since the normal distribution is a continuous distribution we consider few discrete demand points on the demand distribution and calculate the probability values of those demand points. The process of converting probability distribution based demand information into scenario based demand realization is presented in Appendix A.

If number of discrete demand points taken on demand distribution of a product is *m*, after converting this information into scenario based demand realization the total number of scenarios would be equal to

$S = m^{(\text{number of products})}$. Therefore, to keep the problem tractable we consider an example of smaller product family, number of products in the family = 3 (first three products in the case study) for further analysis on the results obtained by both the approaches.

For each product the demand follows a normal probability distribution and the mean and standard deviation is known. Table 6.9 presents the results obtained by both the heuristic approaches for the cases where mean of the demand of each of the product is fixed but with increasing standard deviation. The data used for the results in Table 6.9 is as follows: shortage cost = [\$200, \$100, \$200] and holding cost=\$10, number of demand points on the probability distribution of each product =5

Table 6.9 Results obtained by both the approaches for different cases of demand parameters

Case #	Normal demand vector {(mean, std. dev.), (mean, std. dev.), (mean, std. dev.)}	GAIP Cost value (\$)	PHA Cost value (\$)
1	{(100, 5), (100, 5), (100, 5)}	19726	21428
2	{(100, 10), (100, 10), (100, 10)}	20708	24820
3	{(100, 20), (100, 20), (100, 20)}	23324	28318
4	{(100, 30), (100, 30), (100, 30)}	26895	30425

From Table 6.9, it is obvious that with increase in standard deviation of normal probability distribution of demand, with all other things fixed, the cost value increases, i.e. with increase in variability in demand the stochastic model gives higher cost values. Also, from the table it is clear that GAIP provides slightly better solution than PHA. However, we can see from Table 6.10 that the time taken by GAIP to solve increases exponentially with increase in number of demand points considered on the probability distribution, i.e. with increase in number of scenarios. PHA is a very fast approach and more suitable where number of scenarios considered is large.

Table 6.10* Comparison of computational time for both the approaches

Case #	# of demand points on the probability distribution	GAIP Time in min. (approx.)	PHA Time in min. (approx.)
1	3	61	12
2	5	438	31
3	7	1181	65

*Demand vector for all the three cases = {(100, 10), (100, 10), (100, 10)}

Figure 6.8 shows the effect of variance in the demand and the number of scenarios considered in the stochastic model on the stochastic solution. Figure 6.8 is a plot of total production cost *vs.* number of demand points considered on the demand distribution of each of the product *vs*. the standard deviation in normal demand. The mean of the each of the product is kept fixed to 100 units. We can see from the figure that with increase in number of demand points considered on the probability distribution (more number of scenarios) the stochastic cost value decreases; and with increase in the normal standard deviation the cost value increases. These instances of the example are solved using GAIP approach.

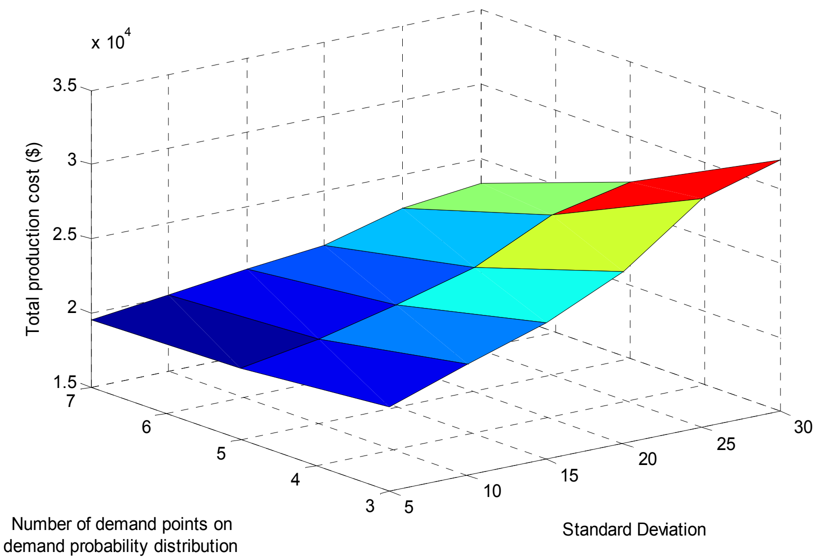


Figure 6.8 The effect of variance in demand and the number scenarios considered in the stochastic model on the stochastic solution

The Figure 6.9 shows that the importance of stochastic model over other models. Figure 6.9 is a plot of objective (cost) value *vs.* standard deviation obtained from using the stochastic model, expected solution model and for the case of perfect information for the example. All the products have same mean values for their demand distribution and standard deviation is increased for all the products. The models are solved using the GAIP approach.

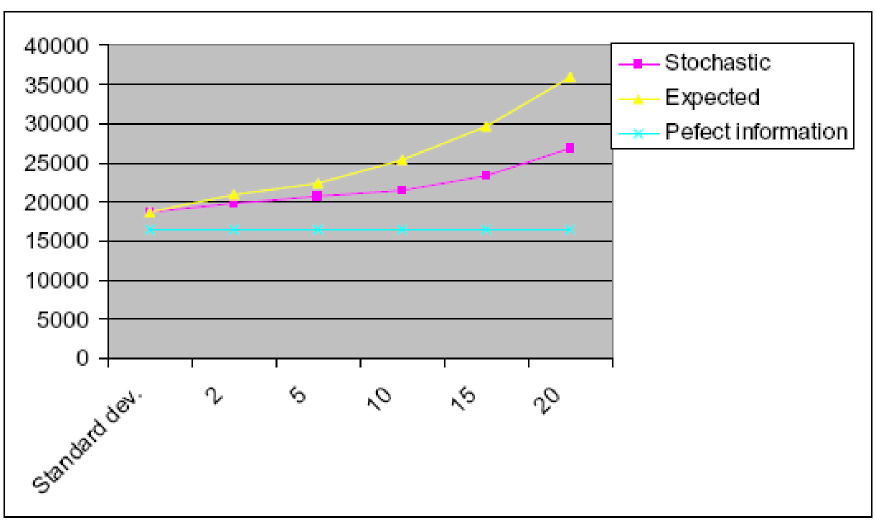


Figure 6.9 Various cost values with increasing standard deviation of the demand distribution

From Figure 6.9 we can conclude that we can use expected value model when the variance in demand is not very high. However, if the variance in demand is significant the stochastic model must be used.

6. 7 Conclusion

The chapter proposes the platform based optimization approach for the economic production of a product family with demand uncertainty. The problem is formulated as a two-stage stochastic integer program with recourse.

Only a very small instance of the problem could be solved by exact approach using OPL 3.5. Therefore, two heuristic methods, that can provide good solutions to large instances of the problem more quickly, are developed. The first heuristic method that combines a genetic search process and integer programming provide a near optimal solution and solves large instance of the problem in a reasonable time, yet this approach takes long time to solve problems with large number of demand scenarios. The second method - pure probability based genetic search heuristic solves the problems with large number of demand scenarios very quickly but with slightly inferior solution quality than the first heuristic approach.

The chapter establishes the use and importance of stochastic model for the platform based production approach especially when the variance in demand is significant. The platform based production approach is explained and illustrated with an example of a family of cordless drills. The research in this chapter provides insight into the effects of demand variance, and various cost components on the optimal platform strategy, and discusses the effectiveness of the heuristics tailored for that application.

CHAPTER 7 - Summary and Future Research

This thesis proposes and establishes the use of platform-based production approach to economically realize mass customization, shorten final assembly time of products leading to increase in customer responsiveness, and economically manage product variety even in uncertain demand environment.

The thesis establishes the motivation, foundation and framework for investigating the proposed research. It provides the background and related literature review on research areas such as platform based production, evolutionary genetic search to solve large scale optimization problems, the touch base on the overall problem environment and assumptions, to establish the context for the readers and foundation leading to the development of three platform based optimization models for the economic production of a given product family and proposing various heuristic solution approaches to solve the problem model efficiently.

The first model considers single platform for the production of a given product family. The model is formulated as a general optimization problem. The problem was solved exactly using OPL 3.5. Also, an evolutionary strategy based heuristic was proposed for the problem. The approach was explained and illustrated with an example of a family of cordless drills presented in Section 3.2. The heuristic provided near optimal solution within 1% of error for most of the problem instances, and in less time when compared to the time taken by the exact approach to get the optimal solution.

The second model is the extension of the first model and considers the production of products based on multiple platforms as opposed to a single platform proposed in first model. The model establishes that using multiple platforms to produce a family of products, given low setup costs, are more economic than using a single platform. Only a very small instance of the problem could be solved exactly using OPL 3.5. Therefore, a genetic algorithm based heuristic was developed that solved the problem very efficiently. The performance of the approach was investigated and the approach was illustrated by applying the heuristic on the case of cordless drills.

The third model is also an extension of the first model and considers the uncertain demand environment. The problem is formulated as a two-stage stochastic integer program with recourse. As only a very small instance of the problem could be solved exactly, two different heuristics – one that combine the genetic algorithm and the integer programming and the other purely probability based multiple population genetic search approach were developed tailored to different instances of the problem.

The heuristics proposed in the thesis provided near optimal solutions and were proved to be very efficient after comparing them with that exact approach by solving the example problem of the drills presented in Section 3.3. Also, the model establishes the importance of the using stochastic programming to efficiently capture the uncertainty in demand especially when the variance in demand is expected to be significant.

Future work in this area includes;

- Proposing multiple platforms for the production of product family in uncertain demand environment.
- Consideration of multi-period demand settings for the products with inventory management policies for platforms and products.
- Consideration of the correlation in demands of the products that can be used to capture *cannibalization* effects or to make the problem more tractable for optimization by reducing the number of independent demand scenarios.

References

1. Allada, V., & Jiang, L., "New modules launch planning for evolving modular product families", *ASME Design Engineering Technology Conf.*, 2002, Paper No. DETC20020DFM-34190.
2. Allada, V., Choudhury, A., Pakala, P. K., Simpson, T. W., Scott, M. J. and Valliyappan, S., "Product platform problem taxonomy: classification and identification of benchmark problems", *ASME Design Engineering Technical Conferences - Design Automation Conference*, Philadelphia, PA, 2006, September 10-13, Paper No. DETC2006/DAC-99569
3. Aytug, H., Khouja, M. and Vergara, F.E., "Use of genetic algorithms to solve production and operations management problems: a review", *International Journal of Production Research,* 2003, 41 (17), pp. 3955–4009.
4. Baldwin, C.Y., and Clark, K.B., "Managing in an age of modularity", *Harvard Business Review,* 1997, 75(5), pp. 84–93.
5. Caffrey, R., Mitchell, G., Wahl, Z. and Zenick, R., "Product Platform Concepts Applied to Small Satellites: A New Multipurpose Radio Concept by AeroAstro Inc.", *16th Annual AIAA/USU Conf. on Small Satellites,* 2002.
6. Dahmus, J.B., Gonzalez–Zugasti, J.P., and Otto, K.N., "Modular product architecture", *Design Studies,* 2001, 22(5), pp. 409– 424.
7. Farrell R. and Simpson T. W., "Improving Commonality in Custom Products Using Product Families", *ASME Design Technical Conferences - Design Automation Conference,* Pittsburgh, *PA,* 2001, September 9-12, Paper No. DETC2001/DAC-21125.
8. Feitzinger, E., and Lee, H.L., "Mass customization at Hewlett–Packard: The power of postponement", *Harvard Business Review,* 1997, 75(1), pp. 116–121.

9. Fujita, K. and Yoshida, H., "Product Variety optimization: simultaneous optimization of module combination and module attributes," *ASME Design Engineering Technical Conferences*, 2001, Paper No. DETC2001/DAC-21058.

10. Fujita, K., Sakaguchi, H. and Akagi, S., "Product variety deployment and its optimization under modular architecture and module communalization," *ASME Design Engineering Technical Conferences*, 1999, Paper No. DETC1999/DFM-8923.

11. Goldberg, E. D., "*Genetic Approach in Search Optimization and Machine Learning*", Addison Wesley Publishing Company Inc., New York, 1989.

12. Gonzalez-Zugasti, J. P., Otto, K. N. and Baker, J. D., "A Method for Architecting Product Platforms", *Research in Engineering Design*, 2000, 12 (2), pp. 61-72.

13. Hernandez, G., Allen, J. K., Mistree, F., "Platform design for customizable products as a problem of access in a geometric space," *Journal of Engineering Optimization*, 2003, 35 (3), pp. 229-254.

14. Hollins, B. and Pugh, S., "*Successful Product Design*", Butterworths, Boston, MA, 1990.

15. Jiao, J. and Zhang, Y., "Product portfolio planning with customer-engineering interaction", *IIE Transactions*, 2005, 37 (9), pp. 801-814.

16. Jose and Tollenaere, "Modular and platform methods for product family design: literature analysis", *Journal of Intelligent Manufacturing*, 2005, 16 (3), pp. 371–390.

17. Kota, S., Sethuraman, K. and Miller, R., "A Metric for Evaluating Design Commonality in Product Families", *ASME Journal of Mechanical Design*, 2000, 122 (4), pp. 403-410.

18. Krishnan, V. and Ulrich, K.T., "Product Development Decisions: A review of the literature," *Management Science*, 2001, 47 (1), pp. 1-21.

19. Lee, H. L. and Tang, C. S., "Modeling the Costs and Benefits of Delayed Product Differentiation", *Management Science*, 1997, 43 (1), pp. 40-53.

20. Lee, H. L., "Effective inventory and service management through product and process redesign", *J. Oper. Res*., 1996, 44 (1), pp. 151–159.

21. Lee, H. L., and C. S. Tang, "Variability reduction through operations reversal" *Management Sci*., 1998, 44 (2), pp. 162–172

22. Li, H. and Azarm, S., "An approach for product line design selection under uncertainty and competition", *ASME Journal of Mechanical Design,* 2002, 124(3), pp. 385–392.

23. Maier J. R.A, and Fadel G. M., "Strategic Decisions in the early stages of product family design," *ASME Design Engineering Technical Conference*, 2001, September 9-12, Pittsburgh, PA, DETC2001/DFM-21200.

24. Martin, M. V. and Ishii, K., "Design for Variety: Development of Complexity Indices and Design Charts", *Advances in Design Automation, Sacramento, CA,* 1997, Paper No. DETC97/DFM-4359.

25. Martin, M.V., and Ishii, K., "Design for variety: Developing standardized and modularized product platform architectures", *Research in Engineering Design,* 2002, 13(4), pp. 213–235.

26. McDuffie, J. P., Sethuraman K., and Fisher M.L., "Product Variety and Manufacturing Performance: Evidence from the International Automotive Assembly Plant Study", *Management Science*, 1996, 42 (3), pp. 350-369.

27. Messac, A., Martinez, M. P., and Simpson, T. W., "Effective product family design using physical programming," *Engineering Optimization*, 2002a, 34 (3), pp. 245-261.

28. Messac, A., Martinez, M.P., & Simpson, T.W., "A penalty function for product family design using physical programming", *ASME Journal of Mechanical Design,* 2002b, 124(2), pp. 164–172.

29. Meyer, M. H. and Lehnerd, A. P., "*The Power of Product Platforms: Building Value and Cost Leadership"*, The Free Press, New York, NY, 1997.

30. Meyer, M. H., "Revitalize Your Product Lines through Continuous Platform Renewal", *Research Technology Management*, 1997, 40 (2), pp. 17-28.

31. Mitchell, M., "An Introduction to Genetic Algorithms (Complex Adaptive Systems)", The MIT Press, Boston, 1998.

32. Moore, W.L., Louviere, J.J., & Verma, R., "Using conjoint analysis to help design product platforms", *Journal of Production and Innovation Management,* 1999, 16(1), pp. 27–39.

33. Naughton, K., Thornton, E., Kerwin, K., & Dawley, H., "Can Honda build a world car"? *Business Week,* 1997, 100(7).

34. Nayak, R. U., Chen, W. and Simpson, T. W., "A Variation- Based Method for Product Family Design", *Engineering Optimization*, 2002, 34 (1), pp. 65-81.

35. Nelson, S. A., II, Parkinson, M. B. and Papalambros, P. Y., "Multicriteria Optimization in Product Platform Design", *ASME Journal of Mechanical Design*, 2001, 123 (2), pp. 199-204.

36. Nemhauser, G. L., and Wolsey, L. A., "*Integer and Combinatorial Optimization*", Wiley-Interscience Series in Discrete Mathematics and Optimization, New York, 1999.

37. Ortega, R. A., Kalyan-Seshu, U., and Bras, B., "A decision support model for the life-cycle design of a family of oil filters," *ASME Design Engineering Technical Conference,* 1999, Paper No. DETC/DAC-8612.

38. Park, J. and Simpson, T. W., "Development of a Production Cost Estimation Framework to Support Product Family Design," *International Journal of Product Research*, 2005, 43 (4), pp. 731-772.

39. Pine, B. J., "*Mass Customization: The New Frontier in Business Competition"*, Harvard Business School Press, Boston, MA, 1993.

40. Rai, R, and Allada, V., "Modular product families design: agent-based Pareto-optimization and quality loss function-based post-optimal analysis," *International Journal Production Research*, 2003, 41 (17), pp. 4075-4098.

41. Robertson, D., and Ulrich, K., "Planning product platforms", *Sloan Management Review,* 1998, 39(4), pp. 19–31.

42. Rothwell, R., and Gardiner, P., "Robustness and product design families", *Design Management: A Handbook of Issues and Methods*, 1990, Cambridge, MA: Basil Blackwell Inc, pp. 279–292.

43. Sabbagh, K., "*Twenty-First Century Jet: The Making and Marketing of the Boeing 777*." New York: Scribner, 1996.

44. Sanderson, S. and Uzumeri, M., "Managing product families: The case of the Sony Walkman", *Research Policy,* 1995, 24 (5), pp. 761-782.

45. Sanderson, S.W., & Uzumeri, M., "*Managing Product Families",* Chicago: Irwin, 1997.

46. Shil, P., and Allada, V., "Evaluating new product platform development projects: A Game theoretic real option approach," *Industrial Engineering Research Conference*, Atlanta, GA, 2005.

47. Siddique, Z., & Rosen, D.W., "Product family configuration reasoning using discrete design spaces", A*SME Design Engineering Technology Conf.*, 2000, Paper No. DETC20000DTM-14666.

48. Simpson, T. W. and D'Souza, B., "A Variable Length Genetic Algorithm for Product Platform Design", *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2002, Atlanta, GA, September 4-6, AIAA, AIAA-2002-5427.

49. Simpson, T. W. and D'Souza, B., "Assessing Variable Levels of Platform Commonality Within a Product Family Using a Multiobjective Genetic Algorithm", *Concurrent Engineering*, 2004, Vol. 12, No. 2, pp. 119-129.

50. Simpson, T. W., "Product Platform Design and Customization: Status and Promise", *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* 2004, 18 (1), pp. 3-20.

51. 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", *Simultaneous Engineering: Methodologies and Applications*, Gordon and Breach Science Publishers, 1999, Amsterdam, The Netherlands, pp. 247-278.

52. Simpson, T. W., Marion, T. J., de Weck, O., Holtta-Otto, K., Kokkolaras, M. and Shooter, S. B., "Platform-Based Design and Development: Current Trends and Needs in Industry", *ASME Design Engineering Technical Conferences - Design Automation Conference*, Philadelphia, PA, 2006, September 10-13, Paper No. DETC2006/DAC-99229.

53. Simpson, T. W., Seepersad, C. C. and Mistree, F., "Balancing Commonality and Performance within the Concurrent Design of Multiple Products in a Product Family", *Concurrent Engineering: Research and Applications*, 2001, 9 (3), pp. 177-190.

54. Simpson, T., Maier, J. and Mistree, F., "A Product Platform Concept Exploration Method for Product Family Design", *ASME Design Engineering Technical Conferences*, Las Vegas, Nevada, 1999, Paper No. DETC99/DTM-8761

55. Spall, J. C., "*Introduction to stochastic search and optimization; estimation, simulation, and control"*, Wiley, Hoboken, NJ, 2003.

56. Sudjitanto, A. and Otto, K. N., "Modularization to Support Multiple Brand Platforms", *ASME Design Engineering Technical Conferences - Design Theory & Methodology Conference, Pittsburgh, PA,* 2001, Paper No. DETC2001/DTM-21695.

57. Swaminathan and Tayur, "Managing Broader Product Lines through Delayed Differentiation Using Vanilla Boxes", *Management Science*, 1998, 44 (12.2), pp. 161-172.

58. Ulrich, K., S. A. Pearson, "Does product design really determine 80% of manufacturing cost?", Working Paper. Sloan School, MIT, MA, 1993.

59. Ulrich, T.K. and Eppinger, S.D., *Product Design and Development*, 2nd edition, McGraw-Hill Inc., New York, 2003.

60. Wilhelm, B., "Platform and Modular Concepts at Volkswagen – Their Effect on the Assembly Process," *Transforming Automobile Assembly: Experience in Automation and Work Organization*, Springer, New York, 1997, pp. 146-156.

61. Wilson, L. O., and Norton, J. A., "Optimal entry timing for a product line extension," *Marketing Science*, 1989, 8(1), pp. 89-112.

62. Womak, J. P. , Jones, D. T., and Roos, D., T h e Machine That Changed the World , , Rawson Associates, New York, 1990.

63. Yano, C. and Dobson, G., "Profit optimizing product line design, selection and pricing with manufacturing cost considerations", *Product Variety Management: Research*

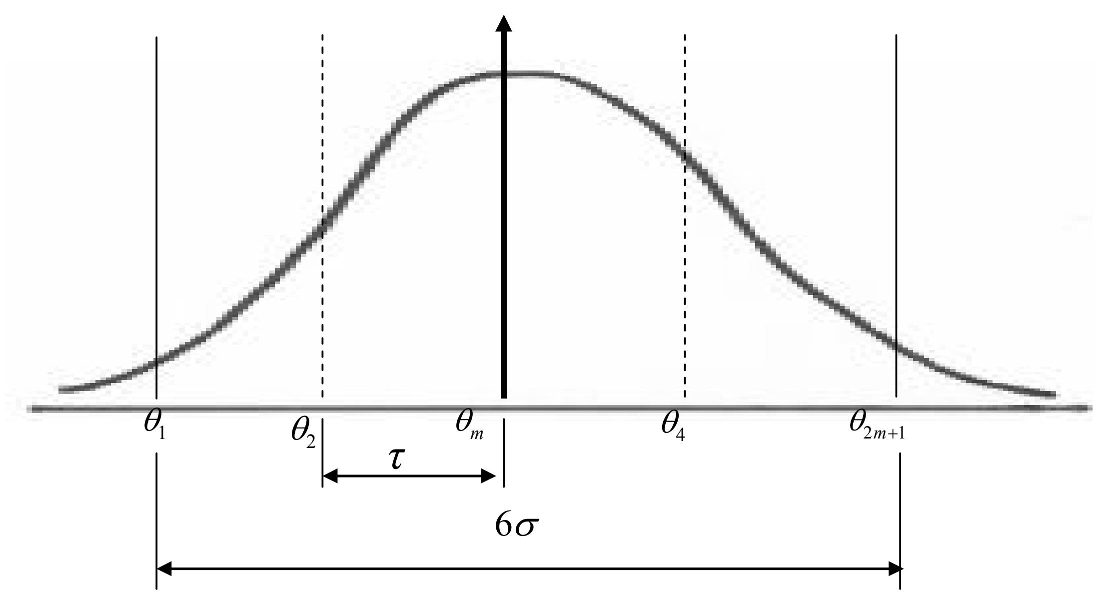
Advances, Ho, T.H. and Tang, C.S. (eds.), Kluwer Academic Publishers, Boston, MA, 1998, pp. 145–176.

Appendix A - Generating all demand scenarios and their probability of occurrences when the normal demand probability distribution parameters for the products are given

The following steps show how from the given normal demand probability distribution of each of the product the demand scenario information (scenarios + probability of occurrence of each of the scenario) of the family of the products is generated.

Step # 1: *Select the discrete demand points on a given normal demand probably distribution of a product in a product family*

The demand of each of product follows normal distribution with mean μ and standard deviation σ. All the discrete demand points (θ_i) considered are from the 6σ width of the normal distribution curve (see figure below). An odd number (2m + 1) points are considered that are equally spaced by τ and symmetrically distributed about the mean.



The value of τ and θ_i s are determined by using the following the relations.

$$\tau = \frac{6\sigma}{2m+1}$$

$$\theta_1 = \mu - (m-1)\times\tau$$

$$\theta_{k+1} = \theta_k + \tau, \quad k = 1,...,2m$$

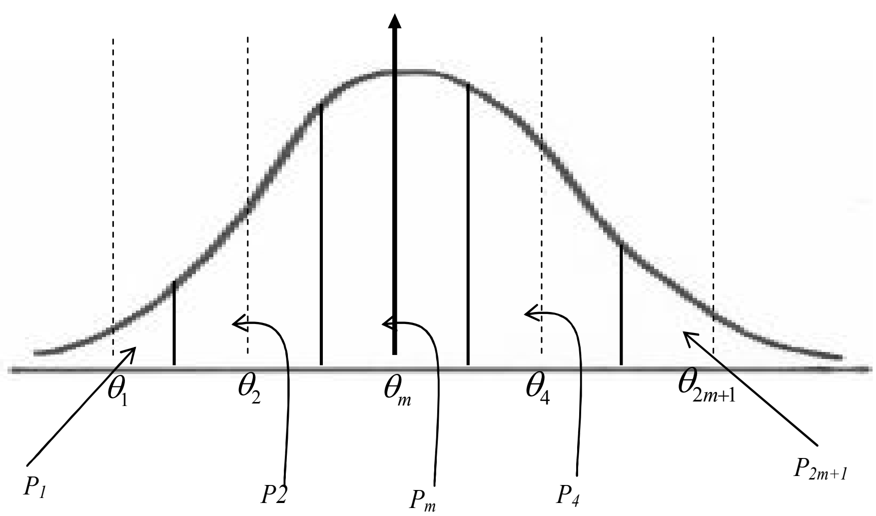
Step # 2: *Calculating the probability value of a discrete demand point on the normal probability curve*

The probability value p_i of a demand point θ_i is calculated by using the following relationships.

$$p_1 = \int_{-\infty}^{\theta_1+\tau/2} f(x)dx \; ; \; p_k = \int_{\theta_k-\tau/2}^{\theta_k+\tau/2} f(x)dx \; ; \; p_{2m+1} = \int_{\theta_{2m+1}-\tau/2}^{+\infty} f(x)dx$$

Where, *f(x)* is the normal probability density function.

In the figure below, the dotted lines show the demand points. The probability value of a demand point is shown by the area enclosed by two solid lines on both sides of the dotted lines except for the end points. For an end point, the probability value is determined by the area which is limited by one solid line at one side and from the other side it extends to infinity.



Step # 3: *Determining the demand scenarios and the probability of occurrence of each of the scenario*

Let θ be a vector of demand points on the demand probability distribution of a product. Then the θ Vector for a product k = [$\theta_1^k \; \theta_2^k \ldots \theta_m^k \ldots \theta_{2m+1}^k$] (determined in Step # 1). The corresponding probability vector for a product $k = [P_1^k \ P_2^k \ \ldots \ P_m^k \ldots \ P_{2m+1}^k]$ (determined in Step # 2).

Then a demand scenario is generated by taking a demand point from demand vector θ of each of the product. All demand scenarios for the products are determined by considering all such combinations. And the probability value of a scenario would be the product of corresponding probability values for the selected demand points in that very scenario vector.

For example, for a case of 3 products and if 5 demand points are considered on the probability distribution of each product, a demand scenario could be $S = [\theta_3^1 \ \theta_2^2 \ \theta_5^3]$ and the probability of occurrence of this scenario would be $p_s = (P_3^1)x(P_2^2)x(P_5^3)$

The total number of scenarios, if the number of demand points considered on each product's probability distribution is same = $(\text{number of demand points})^{\text{number of products}}$