Product portfolio planning with customer-engineering interaction

Zhang, Yiyang

2006

Zhang, Y. Y. (2006). Product portfolio planning with customer-engineering interaction. Doctoral thesis, Nanyang Technological University, Singapore.

https://hdl.handle.net/10356/5266

https://doi.org/10.32657/10356/5266

PRODUCT PORTFOLIO PLANNING WITH CUSTOMER-ENGINEERING INTERACTION

ZHANG YIYANG

SCHOOL OF MECHANICAL AND AEROSPACE ENGINEERING

NANYANG TECHNOLOGICAL UNIVERSITY

2006

PRODUCT PORTFOLIO PLANNING WITH CUSTOMER-ENGINEERING INTERACTION

Zhang Yiyang

School of Mechanical and Aerospace Engineering

A thesis submitted to

Nanyang Technological University

in partial fulfillment of the requirement for the degree of

Doctor of Philosophy

2006

ACKNOWLEDGEMENT

I would like to take this opportunity to express my sincere gratitude to my supervisor, Dr. Roger Jiao, for all his encouragement, help, patience and guidance. I am greatly appreciated for his high expectation and continuous support throughout this research.

The same appreciation goes to my co-supervisor, Prof. Martin Helander, for his advice and inspiration to finish this research.

Special thanks go to Dr. Shaligram Pokharel, Dr. Chen Chun-Hsien, Dr. Arun kumar, and Dr. Yeo Khim Teck for their help and encouragement.

I would also like to express my appreciation to Mrs. Tan Lian Hooi for her assistance during my study in Center for Project Management Advancement (CPMA).

I appreciate all the help from my classmates: Ms. Zhang Lianfeng, Ms. Lim Ching Moi, Mr. You Xiao, and Mr. Ashwin Ravi Ittoo, as well as other friends in China. Their support and friendship encourage me to face the challenge.

Last but not least, I would like to express my love and appreciation to my family for their love, encouragement and support all the time.

ABSTRACT

A critical decision facing many companies across many industries is the selection of an optimal mix of product attributes to offer in the marketplace, which is referred to as product portfolio planning. Product portfolio planning generally involves two stages, namely portfolio identification and optimization. The former aims to capture and understand customer needs effectively and accordingly to transform them into specifications of product offerings. The latter concerns how to determine an optimal mix of these identified offerings to offer in the marketplace.

Current research and industrial practice have mainly focused on the economic justification of a given product portfolio, whereas the portfolio identification issue has received only limited attention. On the other hand, the product portfolio optimization problem has been typically dealt with from a marketing perspective, with the focus on customer concerns – how alternative sets of product attributes and options of attribute levels interact and compete within the target customer segments. From an engineering perspective, the operational implications of product portfolio decisions have been tackled with a primary emphasis on the cost and complexity of interactions among multiple products in a manufacturing environment with increasing variety. Consideration of the customer and engineering interaction in product portfolio planning has become increasingly important, manifested by those efforts in many industries to improve the coordination of marketing, design and manufacturing activities across product and process platforms.

This research develops a systematic framework of product portfolio planning for portfolio decisions while leveraging both customer and engineering concerns. An association

rule mining system is developed to support product portfolio identification through knowledge discovery from past sales and product records. A maximizing shared surplus model, considering customer preferences, choice probabilities, and platform-based product costing, is proposed to address the product portfolio optimization problem. A heuristic genetic algorithm is developed to solve the mixed integer combinatorial optimization problem associated with product portfolio optimization.

To demonstrate the application to the customer-engineering interaction, an associative classification-based recommendation system is developed to support customer decision making in mass customization. A Kansei mining system is developed for customer perception modeling and affective design support. The product portfolio optimization framework is extended to deal with product family configuration design. The results of case studies, along with sensitivity analysis and performance evaluation, suggest the significance of the research problem, as well as the feasibility and potential of the proposed framework.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

This chapter provides an overview of the background knowledge leading to this research. Based on discussion of the research motivation, the research problem is identified as product portfolio planning with customer-engineering interaction, which suggests itself as an important strategy to address the front-end issues of product family development. Accordingly, research objectives and scopes are defined, along with an outline of a technological roadmap for product portfolio planning research.

1.1. Background

Today's consumer markets are changing faster, and consumers are more demanding than ever (Cox and Alm, 1998). It is not uncommon that customers are willing to pay more for those products that meet their unique requirements (Moffat, 1990). Manufacturing companies tend to differentiate their products and provide a huge amount of variety to the marketplace in order to match diverse consumer needs. However, the explosion of product variety unavoidably leads to increased costs in design, production, manufacturing, inventory, and logistics (Da Silveira et al., 2001). In addition, the complexity due to variety proliferation always causes customer confusion (Huffman and Kahn, 1998).

Mass customization has emerged in direct response to these market challenges (Pine, 1993). It aims at satisfying individual customer needs while staying near mass production efficiency (Pine, 1993). It recognizes each customer as an individual and provides each of them with "tailor-made" products (Tseng and Jiao, 1998). As a consequence, customers can no longer be lumped into a homogeneous group in the current marketplace (Hart, 1995).

Mass customization appears as a strategy to differentiate companies in a highly competitive market environment. Compared with mass production, mass customization is characterized by customer-specific product design (Piller et al., 2004), where customers are integrated into design activities to inform designers about what they want. Many researchers have observed that integrating customers into the design and production processes is a promising strategy for companies to react to the growing demand for individualization (Duray and Milligan, 1999; Da Silveira et al., 2001; Tseng and Piller, 2003; Piller et al., 2004). Customer integration enables specific information to be identified, and thus customer needs and desires are defined and translated into concrete product specifications.

In a mass customization system, customer integration occurs at the product definition phase along the entire spectrum of product family development according to the concept of domains (Suh, 2001). As shown in Figure 1-1, product family development in general encompasses three consecutive stages: (1) product definition – mapping of customer needs (CNs) in the customer domain to functional requirements (FRs) in the functional domain; (2) product design – mapping of FRs in the functional domain to design parameters (DPs) in the physical domain; and (3) process design – mapping of DPs in the physical domain to process variables (PVs) in the process domain. The customer, functional, physical and process domains address the customer satisfaction, functionality, technical feasibility, and manufacturability/cost issues associated with the products, respectively (Jiao and Tseng, 1999a). Within the context of mass customization, product design and process design are embodied in the respective product and process platforms. Product definition is characterized by the product portfolio representing the target of mass customization (i.e., the "right" product offerings), which in turn becomes the input to the downstream design activities and is propagated to process design in a coherent fashion. In this regard, a product portfolio represents the functional specification of product families, i.e., the functional view of product and process platforms (Jiao and Tseng, 2004).

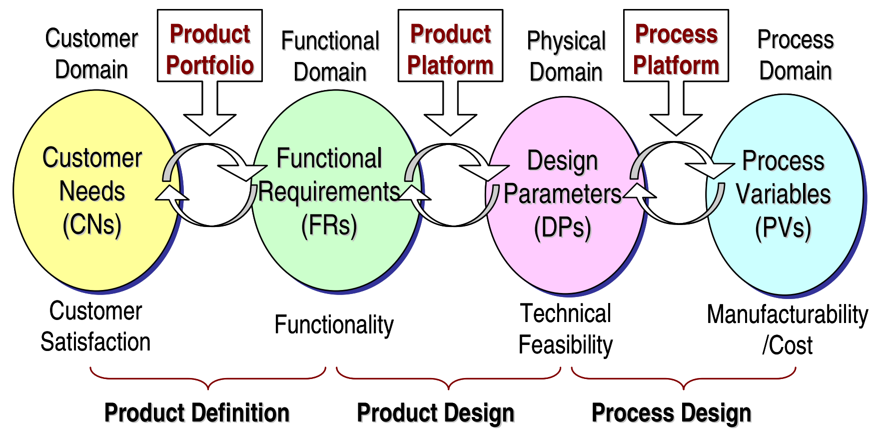


Figure 1-1 Product definition within the spectrum of product family development

1.2. Research Motivation

The product definition phase constitutes the front-end issues of product family development, and is characterized by the product portfolio, it thus must be carefully planned to facilitate downstream activities. Most existing research has emphasized the back-end issues of product family development such as design and manufacturing to enhance the capabilities for mass customization. Over the past decade, a number of strategies and methods have been proposed for developing mass customized products such as product family architecture (Tseng and Jiao, 1996), postponement for supply chain management (Lee and Billington, 1994), design for variety (Ishii and Martin, 1996), and high-variety production planning and control (Hillier, 2000; Jiao et al., 2000; Martinez et al., 2000), to name but a few. The ultimate goal is to fulfill a wide variety of individual needs with reasonably low costs and short lead times. These efforts are mostly geared towards so-called technical variety (Jiao and Tseng, 2000) – diversity of engineering realization in order to achieve various specific customer needs (so-called functional variety) with the focus on the design and manufacturing phases.

On the technical side, the designers always assume that the customer satisfaction increases as a result of good performance of technical capabilities (Jiao et al., 2005). In practice, however, what customers appreciate is not the enhancement of the solution capability per se, but the functionality of the product, i.e., the functional variety. It is not uncommon that some product variants are far more preferred as predicted, while others, although they are equally sound in technical terms, are not favored by customers. In addition, providing a vast variety of options does not always generate customer contentment; instead, it may cause a great deal of confusion and may even turn customers away (Tseng and Piller, 2003). Therefore, it is necessary to examine the underlying interrelationship between customer requirements and product performance, along with the combined effects of multiple product offerings on both customer satisfaction and engineering implications. This suggests that the product portfolio needs to be planned with more consideration of both marketing and engineering decisions and customer perceptions.

1.3. Research Objective and Scope

The primary objective in this research is to develop a systematic framework for product portfolio planning that supports portfolio decisions while leveraging both customer and engineering concerns. Specific problem areas in relation to product portfolio planning are identified as (1) absence of a definite structure for customer requirements; (2) lack of decision support for providing the right product portfolio; (3) inability in adapting to diverse product portfolio planning scenarios; and (4) inability in addressing the granularity issues inherent in product portfolio decisions. Towards this end, necessary tasks are identified as follows.

(1) Investigate the rationale of product portfolio planning and develop a systematic framework of product portfolio planning, in particular,

- Identify the fundamentals of product portfolio planning, including product portfolio identification and product portfolio optimization;
- Analyze the technical challenges and key research issues of product portfolio identification and optimization; and
- Develop appropriate solution strategies for product portfolio identification and optimization.

(2) Develop systematic product portfolio identification methodologies, including:

- Formulate the product portfolio identification problem rigorously;
- Develop systematic procedures and decision-making methods for product portfolio identification based on association rule mining; and
- Validate the system and methods based on the results of case studies.

(3) Develop systematic product portfolio optimization methodologies, including:

- Formulate the product portfolio optimization problem rigorously;
- Develop an optimization model that addresses customer-engineering interaction;
- Develop approaches to customer behavior and engineering analysis in relation to product portfolio optimization;

- Develop a heuristic genetic algorithm to solve the combinatorial optimization problem associated with product portfolio optimization; and
- Validate the model and solution framework based on the results of case studies.

(4) Apply the product portfolio planning framework to address customer-engineering interaction, including:

- Develop a recommendation system to provide support for customer decision-making in mass customization;
- Develop a Kansei mining methodology to support affective design;
- Extend the product portfolio optimization framework to support product family configuration design; and
- Validate these applications based on the results of case studies.

1.4. Organization of the Thesis

Figure 1-2 presents a snapshot of the technological roadmap of this research that encompasses motivation and significance, methodology and solution, application, and validation of the thesis work. The motivation and significance of product portfolio planning research are discussed in Chapters 1 and 2. Chapter 1 discusses the general background of this research with an outline of a holistic view of platform-based product development and product family design. Chapter 2 provides a comprehensive review of the state-of-the-art research in the field. The review is organized according to various topics in relation to product families, including platform-based product development, product family design, manufacturing and production for product families, customer integration for product families, economic justification, customer needs elicitation and requirement analysis, optimal product design, product positioning, and product line design. Chapter 3 presents the fundamental issues underlying product portfolio planning with customer-engineering interaction. Discussed in detail are the portfolio strategy, technical implications and key challenges, as well as the respective solution strategies.

Product portfolio planning implies two fundamental elements, namely product portfolio identification and product portfolio optimization. Chapters 4 and 5 emphasize these two topics, respectively. In Chapter 4, an association rule mining system (ARMS) is developed to support product portfolio identification. The mapping mechanism between the customer and functional domains is incarnated in the association rules. The ARMS architecture and implementation issues are elaborated, along with a case study in a consumer electronics company for generating the portfolio of vibration motor products.

Chapter 5 reports the development of a product portfolio optimization model and the corresponding solution framework. To leverage both the customer and engineering concerns, a maximizing shared surplus model, considering customer preferences, choice probabilities and platform-based product costing, is proposed to address the product portfolio optimization problem. A heuristic genetic algorithm is developed to solve the mixed integer combinatorial optimization problem inherent in product portfolio optimization. A case study of notebook computer portfolio optimization is presented to illustrate the feasibility and potential of the proposed framework.

Chapter 6 is devoted to the applications of the proposed product portfolio planning framework to deal with customer-engineering interaction. Three application areas are demonstrated in relation to customer-engineering interaction, including customer decision-making in mass customization, affective design, and product family configuration design. An associative classification-based recommendation system is developed to facilitate customer decision making in an online mass customization scenario. A Kansei mining system is developed to capture customers' perceptions and to provide affective design support. The product portfolio optimization framework is extended to tackle product family configuration design. A generic genetic algorithm is formulated to solve the design evaluation problem, where a generic encoding scheme is applied to adapt to diverse product family configuration scenarios, and a hybrid constraint-handling strategy is developed to cope with complex constraints involved in product family configuration design.

The last chapter, Chapter 7, summarizes the achievements in addressing the research objectives and tasks. A critical assessment is given to highlight the limitations and possible improvements of the thesis work, along with recommendations for future work.

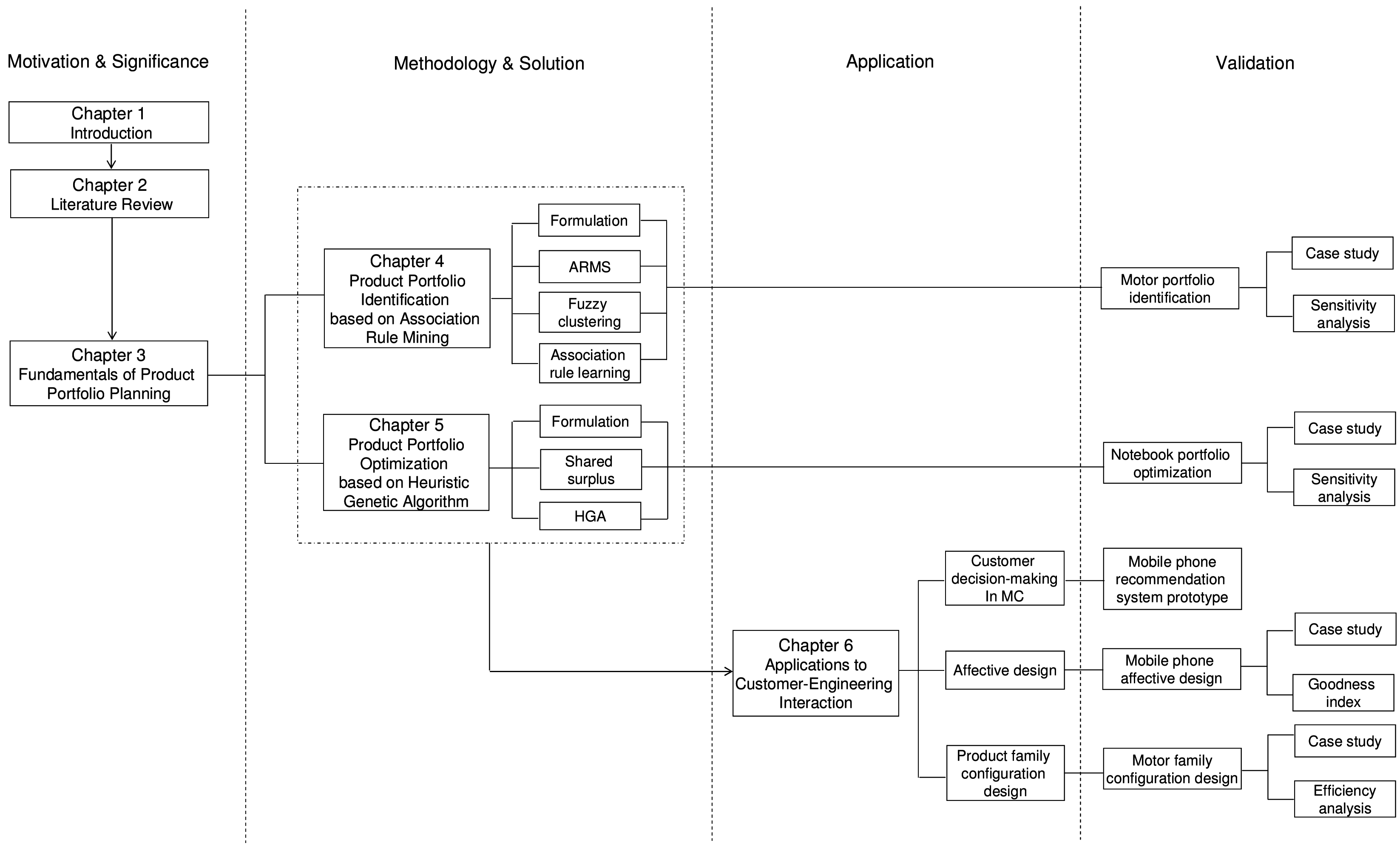


Figure 1-2 Organization of the thesis

CHAPTER 2
LITERATURE REVIEW

Product family design and platform-based product development have received much attention over the last decade. This chapter provides a review of the state-of-the-art research in this field. Major challenges and future research directions are also discussed. It highlights the motivation to carry out an in-depth study on product portfolio planning.

2.1. Platform-based Product Development

Platform-based product development has been well recognized as an effective means to achieve the economy of scale in order to accommodate increasing product variety across diverse market niches (Meyer and Utterback, 1993; Sundgren, 1999). A sizeable body of research on platform-based product development has been reported over the last decade (Simpson, 2004).

2.1.1. Product Family

Streams of individual products generated by firms may be thought of as evolving families of products (Meyer and Utterback, 1993). A product family refers to a set of similar products that are derived from a common platform and yet possess specific features/functionality to meet particular customer requirements (Meyer and Lehnerd, 1997). Each individual product within a product family, i.e., a family member, is called a product variant or instance. While a product family targets a certain market segment, each product variant is developed to address a specific subset of customer needs of the market segment.

All product variants share some common structures and product technologies, which form the platform of the product family (Erens and Verhulst, 1997).

The interpretation of product families depends on different perspectives. From the marketing and sales perspective, the functional structure of product families exhibits a firm's product line or product portfolio and thus is characterized by various sets of functional features for different customer groups (Agard and Kusiak, 2004). The engineering view of product families embodies different product technologies and associated manufacturability and is thereby characterized by various design parameters, components, and assembly structures (Simpson, 2004; De Lit and Delchambre, 2003).

2.1.2. Product Platform

Product platforms have been defined diversely, ranging from being general and abstract (Robertson and Ulrich, 1998) to being industry and product specific (Sanderson and Uzumeri, 1995). In addition, the meaning of platform differs in scope. Some definitions and descriptions focus mainly on the product or artifact itself (Meyer and Utterack, 1993; McGrath, 1995), whereas others try to explore the platform concept in terms of a firm's value chain (Sawhney, 1998).

There are two streams of research prevailing in the field of developing product platforms. One perspective refers to a platform as a physical one, namely a collection of "elements" shared by several products. Accordingly, the major concern is how to identify the common denominators for a range of products (Wilhelm, 1997). The effort is geared towards the extraction of those common product elements, features, and/or subsystems that are stable and well understood, so as to provide a basis for introducing value-added differentiating features (Moore et al., 1999). Meyer and Lehnerd's work (1997) is the representative of another dominating perspective to product platform. They define a product platform as "a set of subsystems and interfaces developed to form a common structure from which a stream of derivative products can be efficiently developed and produced" (Meyer and Lehnerd, 1997). The major issue is to exploit the shared logic and cohesive architecture underlying a product platform. McGrath (1995) defines product platform as a collection of the common elements, especially the underlying core technology, implemented across a range of products. Robertson and Ulrich (1998) define a platform as the collection of assets that are shared by a set of products. The assets include components, processes, knowledge, as well as people and relationships.

Baldwin and Clark (2000) define three aspects of a product platform: (1) its modular architecture, (2) the interfaces, and (3) the standards that provide design rules to which the modules must conform. To facilitate platform-based product family development, interface management is reported as a distinct process of defining the physical interfaces between subsystems (Sundgren, 1999). Zamirowski and Otto (1999) discern three types of product platforms: modular platforms, scalable platforms, and generational platforms. A modular platform is used to create variants through configuration of existing modules (Meyer and Lehnerd, 1997). A scalable platform facilitates the differentiation of variants that possess the same function with varying capacities. A generational platform leverages product life cycles for rapid next generation development (Martin and Ishii, 2002). One endeavor towards product platform development is to design product families in the way of "stretching" or "scaling" (Rothwell and Gardiner, 1990).

2.1.3. Product Architecture

The concept of architecture, with respect to product design, is synonymous with the layout, configuration, or topology of functions and their embodiment (van Wie et al., 2003). Product architecture can be defined as the way in which the functional elements of a product are arranged into physical units and the way in which these units interact (Ulrich and Eppinger, 1995). Fujita and Yoshida (2004) point out one important characteristic to discern the architecture of a family of products from that of a single product – the simultaneous handling of multiple products. Erens and Verhulst (1997) consider the functional and physical architectures for product families and describe them using a package of single product models. Yu et al. (1999) approach product architectures from a functional perspective by defining the architecture based on customer demands. Ulrich (1995) discusses the relationship between product architectures and managerial problems related to product strategies.

The typology of product architectures suggests that the architecture can be either integral or modular (Muffatto and Roveda, 2002). Modularity has been well studied from many perspectives (Fixson, 2002; Bi and Zhang, 2001). Ulrich and Tung (1991) define five categories of modularity, i.e., component swapping, component sharing, fabricate-to-fit, bus and sectional modularity. Pine (1993) adds a sixth: mix modularity, which is frequently encountered in the painting and chemical industries. While most extensions (Kusiak and Huang, 1996; Du et al., 2001) are built upon these basic modularity types, the current practice mostly refers to the product architecture as physical structures in terms of physical parts or components (Henderson and Clark, 1990).

While modularity deals with the mapping from functions to components, integrality involves standardization and decoupling of the interfaces between components (Ulrich and Eppinger, 1995). Robertson and Ulrich (1998) observe that increasing modularity with proper integrity is conducive to the management of tradeoff between distinctiveness and commonality in product architectures. Sosa et al. (2003) observe the importance of integrality and modularity in design team interactions and introduce a method of identifying whether a system is modular or integral based on analysis of component interactions using a design structure matrix (DSM). Fixson (2002) constructs a DSM to analyze the total number of functions that components under consideration provide on the other, based on which modular and integral architectures are identified. Whitney (2003) studies total modularity and interfaces in the context of design economy. Cutherell (1996) finds that integral architectures are often driven by product performance or cost, while modular architectures are driven by variety, product change, engineering standards, and service requirements.

Jiao and Tseng (1999a) assert that a product family architecture involves systematic planning of modularity and commonality in terms of building blocks and their configuration structures across the functional, technical and structural views. Zamirowski and Otto (1999) point out the necessity to develop the product architecture and platform by synchronizing multiple views such as those from customer needs, functional structures and physical architectures. The leveraging of modularity and commonality in product family architecture development is also supported by Siddique et al. (1998). Muffatto and Roveda (2002) study multiple aspects of product architectures including functions, requirements, technological solutions, product concepts, product strategies and platforms, as well as production and assemblies. To address the question of how differences in the product architecture affect resource consumption during the design phase, Eppinger et al. (1994) link the task structure of the design process to the product architecture.

2.2. Product Family Design

Corresponding to the scalable and modular product platforms, there are two types of approaches to product family design. One common approach is called scalable (namely parametric) product family design, whereby scaling variables are used to ‘‘stretch’’ or ‘‘shrink’’ the product platform in one or more dimensions to satisfy a variety of customer needs. The other approach is referred to as configurational product family design, which aims to develop a modular product platform, from which product family members are derived by adding, substituting, and/or removing one or more functional modules (Du et al., 2001).

2.2.1. Scalable Product Family Design

Scalable product family design involves two basic tasks (Simpson, 2004). The first one is platform selection – to determine which design parameters take common values. While many existing methods assume that the platform architecture is known a priori (Fujita et al., 1999), some approaches determine platform variables along with scalable variables during optimization (Akundi et al., 2005; Dai and Scott, 2004). The subsequent task is to determine the optimal values of common and distinctive variables by satisfying performance and economic requirements. Most approaches consider only a single product platform, where each platform variable is shared across the entire product family. This strategy excels in computational simplicity, but may lead to a situation where some low-end products may be over-designed and certain high-end products may be under-designed (Dai and Scott, 2004). The other strategy is to consider multiple product platforms in product family design, such that design variables can be shared by any subset of product variants within the product family (de Weck et al., 2003). Multiple-platform design enhances exploration of the solution space, whereas sacrificing the computational efficiency (Seepersad et al., 2002).

2.2.2. Configurational Product Family Design

The configurational approach to product family design is also frequently called module-based product family design (Simpson, 2004). It is based on the development of modular product architectures. As defined by Ulrich and Tung (1991), a modular product architecture involves one-to-one mappings from functional elements in the function structure to the physical components of a product, where decoupled interfaces between components can be specified. Ulrich (1995) points out that the modular product architecture allows each functional element of the product to be changed independently by changing only the corresponding component. This is advantageous to produce custom-built products from standard models. It also makes standardization possible, which is essential to achieve the economy of scale; therefore, using modular product architectures, variety can be created by configuring existing building blocks. Salient issues regarding configurational product family design include module identification, interface standardization, and architecture embodiment as discussed next.

Erlandsson et al. (1992) develop a method with three major steps to help identify product modules. In their method, the right product specification is attained by adopting quality function deployment (QFD). Module creation, interface analysis and module configuration are carried out by creating different modular structures according to the QFD matrix (i.e., the house of quality). Erixon and Ostgren (1993) extend this method by applying the QFD matrix to modular analysis and coin it as modular function deployment (MFD) with focus on the evaluation of module integration. Yu et al. (2003) apply the DSM as a tool to identify highly interactive groups of product elements and to cluster them into modules. Hölttä and Salonen (2003) compare three modularization methods using commercial products. They reveal that the MFD method is the least repeatable, whereas the computerized DSM method is the most repeatable, and the heuristic approach falls in between. Malmström and Malmqvist (1998) integrate the DSM and MFD methods to tackle both technical and economical aspects in the early stages of product architecture development. Stone et al. (2000) formulate a set of heuristics for grouping functions to form a module. Hölttä et al. (2003) develop a five-step algorithm for grouping and creating a dendrogram for finding common modules across products for platforming a product family. Salhieh and Kamrani (1999) employ a clustering technique for identifying design modules. Otto et al. (2000) propose a framework for architecting a family of products that share interchangeable modules. They define a modularity matrix for one family of products from a manufacturer, allowing commonalties to be easily identified. Gershenson et al. (2003) provide an extensive comparison of several DSM-based methods for identifying modular architectures.

2.2.3. Metrics for Product Family Design

Product family design essentially entails a type of multi-objective optimization (Simpson et al., 2005). In many cases, such multiple criteria decision-making, given a number of alternatives at different levels of abstraction of the product architecture, requires tradeoffs between three criteria: cost, revenue and performance. In addition, it needs to weigh the revenue from product cannibalization by commonality with respect to the cost savings from commonality (Robertson and Ulrich, 1998).

(1) Modularity. Prasad (1998) studies the product and process complexity associated with design for variety, highlighting the importance of determining the right amount of

decomposition. To quantify such a granularity paradox, a measure of communication effort is introduced in order to achieve an optimal balance. Gershenson et al. (2003) develop a measure of relative modularity for modular product design. Mikkola and Gassmann (2003) assume that the degree of modularity in a given product architecture is constrained by the composition of its components. Allen and Carlson-Skalak (1998) introduce some measures of modularity for conceptual design. Ulrich (1995) simply defines the function-to-component ratio for each product as a modularity metric. Hölttä and Salonen (2003) propose a measure of modularity based on singular value decomposition of the binary DSM. Guo and Gershenson (2004) develop a metric to measure product modularity using a component-to-component connectivity matrix. Siddique and Rosen (2001) account for both functional and form issues in their partitioning method that involves combinatorics. Stone et al. (2000) develop product family and customer needs ratings for modules.

(2) Commonality. Kota et al. (2000) develop a measure that captures the level of commonality in a product family. With application to automotive underbodies, Siddique et al. (1998) propose to measure component commonality and connection commonality in order to capture characteristics of platform commonality and product variety. Maupin and Stauffer (2000) take into account simplicity, direct costs, and delayed differentiation for commonality metrics. Emphasizing on component sharing, Ramdas et al. (2003) present a methodology for determining which version of a set of related components should be offered to optimally support a defined finished product portfolio. In the work of Fellini et al. (2002), an optimal design problem is formulated as the maximization of commonality by choosing the product components to be shared without exceeding a user-specified performance loss tolerance and subject to different levels of performance losses. McAdams and Wood (2002) develop a

quantitative metric for design-by-analogy based on the functional similarity of products. Thevenot and Simpson (2005) compare various commonality indices for assessing product families, including the Degree of Commonality Index (Collier, 1981), Total Constant Commonality Index (Wacker and Trelevan, 1986), Product Line Commonality Index (Kota et al., 2000), Percent Commonality Index (Siddique et al., 1998), Commonality Index (Martin and Ishii, 1997), and Component Part Commonality Index (Jiao and Tseng, 2000).

(3) Variety/Distinctiveness. Martin and Ishii (1997) quantify the costs of providing variety in order to quantitatively guide designers in developing products that incur minimum variety costs. Through commonality analysis, van Wie et al. (2006) study how differences between platform elements and differentiating elements are evidenced in the product layout or configuration. Simpson and D'Souza (2004) introduce a genetic algorithm-based approach to product family design that balances the commonality of the products in the family with the individual performance (i.e., distinctiveness) of each product in the family. Dobrescu and Reich (2003) propose a variety index and a standardization index that resemble the commonality indices of Martin and Ishii (2002).

(4) Cost. Kim and Chhajed (2001) develop an economic model that considers a market consisting of a high segment and a low segment. They determine that large commonality decreases production costs but makes the products more indistinguishable from one another, which makes the product more desirable for the low segment but less desirable for the high segment. Fisher et al. (1999) present an analytic model of component sharing based on empirical testing of varying practice of component sharing for automotive braking systems. Fujita and Yoshida (2004) develop a monotonic cost model for the assessment of benefits of commonality. Gonzalez-Zugasti et al. (2000) propose a methodology to design product

platforms and variants with consideration of technical performance requirements and product family costs. Fixson (2005) outlines a roadmap for product architecture costing from a product life cycle perspective. Park and Simpson (2005) examine the effects of commonality decisions on individual costs based on activity-based costing. To link modularity and the cost, Fixson (2002) develops a multi-dimensional product architecture description method that considers the level of function-component allocation, interface intensity, interface reversibility, and interface standardization. Siddique and Repphun (2001) assess the cost implications of product architectural decisions when product architectures allow sharing of parts, modules, or components of a product across product families. The savings from the reuse of designs are shown to affect both development cost and time (Siddique, 2001).

(5) Profit/Valuation. Numerous methods dealing with optimal design use various objectives originated from the profit or expected revenue (Fujita and Yoshida, 2004; Nelson et al., 2001). Many studies have revealed that such a profit measure based on the dollar value is unrealistic in most cases (Tarasewich and Nair, 2001). As such, researchers have been developing various instruments to improve the measurement of profit performance. Balakrishnan and Jacob (1996) introduce share of choices as the objective. Michalek et al. (2005) formulate the evaluation problem as profit maximization by minimizing the technical performance deviation. De Weck et al. (2003) propose to optimize product platform design by maximizing overall product family profitability and reducing the development time and cost. Typical approaches to estimate costs and values coincide with the traditional principle of capital budgeting that is based on discounted cash flows (DCF) analysis. When dealing with numerous options associated with product family design, the DCF approach tends to underestimate the upside potentials to a design project from management flexibility (Kogut

and Kulatilaka, 1994). Real options have been applied to value specific aspects of product development, such as design modularity (Baldwin and Clark, 2000). Otto et al. (2003) explore the real options concept for determining proper levels of independent product architectural attributes.

(6) Platform-related Metrics. Meyer and Lehnerd (1997) develop two platform related measures, named platform efficiency and platform effectiveness, for evaluating the performance of product families. Focusing on the generational aspect of product platforms, Martin and Ishii (2002) develop two indices, called Generational Variety Index and Coupling Index, to measure a product's architecture. De Weck et al. (2003) adopt a market segment model using the sales volume, the price and the competing product alternatives for product family and platform portfolio optimization. Jiao and Tseng (2004) develop a design customizability index and a process customizability index for evaluating the cost effectiveness of a design to be customized in order to meet individual customer needs. Zha et al. (2004) introduce two metrics, market efficiency and investment efficiency, for the evaluation and selection of product design for mass customization.

2.2.4. Product Family Modeling

Baldwin and Clark (2000) develop a discipline-independent data model to provide constructs for modeling products with optional contents. Felfernig et al. (2001) apply the unified modeling language to the modeling of configuration knowledge bases for mass customizable products. The initiative of Product Family Classification Tree emphasizes the classification of end-products and/or modules from a functional viewpoint (Bei and MacCallum, 1995). To facilitate representations from multiple perspectives, Generic Product Modeling is advocated to represent product families from both commercial and assembly views (Wortmann et al., 1997). Siddique and Rosen (1999) develop a graph grammar approach to product platform design. Set-based modeling is an attempt to formalize the representation of product platform design and manufacturing processes (Finch, 1999). Männistö (2000) studies the conceptual modeling of product families, with particular emphasis on the problems related to the evolution of product family descriptions and the product individuals created according to them.

Van Wie et al. (2003) consider product architecture representation as organizing a deluge of information in terms of both function and form. To model product family configuration, Zhang et al. (2005) propose to organize and manage product knowledge through a knowledge component that includes configuration rules and constraints. Bohm and Stone (2004) investigate the representation of functionality for supporting reuse. Sharman and Yassine (2004) study some forms of abstraction for describing product architectures, including DSM, molecular diagrams, and visibility-dependency signature diagrams. Costa and Young (2001) introduce a product range model (i.e., product families) for information modeling of variant and adaptive design. Tiihonen et al. (1998) develop a method of managing and modeling a product family as a configurable product, which is based on the conceptualization of components, attributes, resources, ports, contexts, functions and constraints. Jiao et al. (1998) observe different data types underlying product families that involve product-to-product, product-to-family and family-to-family relationships. To characterize variety and its derivation, Jiao et al. (2000) propose a generic variety structure consisting of common product structures, variety parameters, and configuration constraints.

2.2.5. Product Family Design Support Systems

Kusiak and Huang (1996) and O'Grady and Liang (1998) put forth design with modules that centers around module selection. Liang and O'Grady (2000) focus on a particular design environment where modules may be available from one or more geographically dispersed sources, and where data concerning the modules may be in a multitude of databases scattered across the globe. Huang and Liang (2001) develop a formalism for design with modules, such that customer requirements are met using modules from suppliers geographically separated through diverse computer platforms.

Online product configurators have recently received much attention to enable customers to interactively specify and adapt a product according to their individual preferences (Sabin and Weigel, 1998). Bramham and MacCarthy (2003) examine the empirical evidence of available configurators in terms of matching configurator attributes against business strategies. Hvam (2004) reviews the design and implementation of product configuration systems from the viewpoint of industrial applications. Simpson et al. (2003) investigate a framework for web-based platform customization. Common configuration systems for product families necessitate product-specific knowledge and often overstrain customers (Blecker et al., 2004). Advisory systems are thus advocated to guide customers according to their profile and requirements through a personalized configuration process ending with the generation of product variants that better fulfill the real customer needs (Blecker et al., 2004). Ardissono et al. (2003) report on an EU-funded project, CAWICOMS Workbench, which aims at next generation Web-based applications that support distributed configuration of products and services within a supply chain.

2.3. Manufacturing and Production for Product Families

While seeking technical solutions is the major concern in design, it is at the production stage that product costs are actually committed, and product quality and lead times are determined per se. For a given design, the actual cost depends on how production is planned and to what extent the economy of scale can be realized within the existing manufacturing capabilities. This implies that the claimed rationale of product family design can only be fulfilled at the production stage (Jiao and Tseng, 2004).

The traditional approach to deal with a large number of variants associated with product families is to treat each product as an individual bill-of-materials (BOM), which however leads to a data explosion problem (Olsen and Sætre, 1996). To overcome the limitations of traditional BOMs in handling a large number of variants, the generic BOM (GBOM) concept is developed by van Veen (1992). The GBOM defines a generic product as a set of variants that can be identified through specifying alternative values for a set of parameters (Hegge and Wortmann, 1991). The generic bill-of-materials-and-operations is put forth by Jiao et al. (2000) by unifying BOMs and routings to accommodate large numbers of product and process variants. For multi-product and multi-process production systems, Aydiny and Gugor (2005) develop a relational database approach to generating BOMs and executing MRP.

Meyer and Lehnerd (1997) expand the common definition of a platform to include possible commonality in processes and production. In particular from the production and assembly perspectives, a platform implies a focus on commonality of production tools, machines and assembly lines (Sanchez, 1994). As a consequence, some companies in the automotive industry have considered it more interesting to define a platform on a manufacturing-assembly basis rather than on a product development basis, so as to better exploit commonality among production process (Wilhelm, 1997).

A benefit of designing product families comes from a reduction of components in inventory and component handling, reduction of component types and manufacturing processes, and increased production volumes (Fisher et al., 1999). However, sharing components in a product family may lead to a lack of distinctiveness, and shared components in one product often exceed the requirements of other products, which causes additional production costs (Krishnan and Gupta, 2001). Nobelius and Sundgren (2002) point out that the potential managerial difficulties associated with the part sharing process involve organizational, strategic, technology and cost related issues. Tsubone et al. (1994) study the relationship between component part commonality and manufacturing flexibility. Siddique et al. (1998) and Wilhelm (1997) demonstrate that the level in the product hierarchy at which commonality is pursued varies with respect to the deployment of production processes.

2.4. Customer Integration for Product Families

The driving force behind product family design and development is the enterprises' positioning of customers at the center of value creation and involving customers into the product fulfillment process. On the technical side, designers have always assumed that customers' satisfaction with the designed product is sufficiently high as long as the product meets the prescribed technical specifications; however, what customers appreciate is not the enhancement of the solution capability but the functionality of the product. This means that the traditional dimensions of customer satisfaction may deserve scrutiny, for example, identifying those product characteristics that cause different degrees of satisfaction among customers; understanding the interrelation between the buying process and product satisfaction; determining the optimal amount of customization and customer integration; and justifying an appropriate number of choices from the customers' perspective.

Equally important are customers' decision-making processes. In the end, providing decision support to customers is important. This coincides with consumer behavior in business systems based on customer involvement in the product customization process (Huffman and Kahn, 1998). While most platform-based customization approaches implemented in practice are based on offering a huge number of variety and choices, the perception of choice and the joy or burden of configuration experienced by customers are not well understood. Many questions are pending. For example, what are the incentives for integrating customers into value creation? What factors drive customers to interact with a configurator? How many variants should be explored and changed before making a final decision? Are there any specific patterns that customers follow when interacting with a platform-based product development system? And how can various players (customers, designers, suppliers, production engineers, etc.) communicate well within the same platform of product family design? Toward this end, product family development needs to be incorporated with more marketing and engineering decisions (Michalek et al., 2005), as well as customer perceptions (Blecker et al., 2004).

2.5. Economic Justification

Product family design and development is associated with new cost and profit structures that can be coined as "economies of scale and scope". Current research on the economic and performance evaluation of product families is dominated by empirical studies, ad hoc samples, or broad approaches based on cost accounting. Traditional cost accounting by allocating fixed costs and variable costs across multiple products may produce distorted cost-carrying figures due to possible sunk costs associated with investment into product and process platforms (Jiao and Zhang, 2005). Safizadeh et al. (2000) derive results from an empirical study of 142 manufacturing plants, such that plants that provide a high degree of customization incur high cost structures; however, when controlling for production processes the tradeoff disappears. This means once a company has defined its product range along with an appropriate production process, platform-based customization that falls within the range offered does not cost any extra.

The economic justification of product families requires the identification of proper measures and performance indicators to characterize different outcomes of a product customization system. This task is imperative because the current accounting systems are not designed for assessing the true economical benefits from the total value chain point of view. Even if the focus is shifted from cost control to value creation, existing accounting and control systems are mostly dominated by the practice of product costing. Savings and additional costs resulting from different degrees of interaction with the customers are not covered by most industrial accounting systems. Activity-based costing and the balanced score card approaches may provide initial solutions; however, approved ratios for calculating the value of customer relationships are still missing; nor are parameters for evaluating the extent of the market research information gained by aggregated customer knowledge. Moreover, the value contribution should be evaluated from the customers' perspective. Only if the increment in the customer-perceived value or utility suffices enough can product customization become a mass phenomenon.

2.6. Customer Needs Elicitation and Requirement Analysis

Approaches to defining product specifications by capturing, analyzing, understanding, and projecting customer requirements, sometimes called the Voice of the Customer (VoC), have received a significant amount of interest in recent years (McKay et al., 2001). A method used for transforming the VoC to product specifications has been developed by Shoji et al. (1993), in which semantic methods, such as the Kawakita Jiro (KJ) method (i.e., affinity diagram) and multi-pickup method (MPM), are applied as the basis for discovering underlying facts from affective language. Kano et al. (1984) propose a model to categorize different types of customer requirements for product definition.

In this regard, market researchers have emphasized customer profiling by applying regression analysis to compare customer characteristics and to determine their overall ranking in contribution towards profitability (Jenkins, 1995). Traditionally, market analysis techniques are adopted for investigating customers' responses to design options. For example, conjoint analysis is widely used to measure preferences for different product profiles and to build market simulation models (Green and DeSarbo, 1978). Louviere et al. (1990) use discrete choice experiments to predict customer choices pertaining to design options. Turksen and Willson (1993) employ fuzzy systems to interpret the linguistic meaning regarding customer preferences as an alternative to conjoint analysis. Others have taken a qualitative approach and used focus groups to provide a reality check on the usefulness of a new product design (LaChance-Porter, 1993). Similar techniques include one-on-one interviews and similarity-dissimilarity attribute rankings (Griffin and Hauser, 1992). While these types of methods are helpful for discovering the VoC, it is still difficult to obtain design requirement information because marketing folks do not know what engineers need to know. It is difficult to apply the VoC alone to achieve a synergy of marketing and engineering concerns in developing product specifications (Veryzer, 1993).

As a structured questioning methodology built upon Kelly's repertory grid technique (Kelly, 1955), the laddering technique has been widely used to transform customers' psychological factors into useful inputs for design applications (Rugg and McGeorge, 1995). Many methods and tools in the field of knowledge acquisition, such as observation, self-report (Cortazzi and Roote, 1975), interview, protocol, ethnographic methods (Mead, 1928), and sorting techniques (Shaw, 1980), have some applicability in requirement elicitation for product development (Shaw and Gaines, 1996). Maiden and Rugg (1996) propose a framework called acquisition of requirements (ACRE) to assist practitioners in understanding the strengths and weaknesses of each of the methods for requirement elicitation. Chen and his co-authors propose an integrated approach to the elicitation of customer requirements by combining picture sorts, fuzzy evaluation, laddering, and neural network techniques (Chen and Occeña, 1995; Chen et al., 2000, 2002; Yan et al., 2001, 2002).

From an engineering design perspective, Hauge and Stauffer (1993) develop a taxonomy of product requirements to assist in traditional qualitative market research. To elicit knowledge from customers (ELK), the taxonomy of customer requirements is deployed as an initial concept graph structure in the methodology for question probe – a method used in the development of expert systems. While ELK aims at making customer information more useful to the designer, the taxonomy developed for ELK is too general to be a domain independent framework (Tseng and Jiao, 1998). McAdams et al. (1999) propose a matrix approach to the identification of relationships between product functions and customer needs. A key component of Quality Function Deployment (QFD; Clausing, 1994) is the customer requirements frame to aid the designer's view in defining product specifications. Researchers at IBM have applied structured brainstorming techniques to build customer requirements into the QFD process (Byrne and Barlow, 1993). While QFD excels in converting customer information to design requirements, it is limited as a means of actually discovering the VoC (Hauge and Stauffer, 1993). Olewnik and Lewis (2005) posit that 'the use of QFD as a quantitative decision support tool in engineering design is potentially flawed'. To empower QFD with market aspects, Fung and Popplewell (1995) propose to pre-process the VoC prior to its being entered as customer attributes into the House of Quality (HoQ). In this process, the VoC is categorized using an affinity diagram (KJ method). Fung et al. (1998) further adopt the Analytic Hierarchy Process (AHP; Saaty, 1980) to analyze and prioritize customer requirements. Fung et al. (2002) extend their QFD-based customer requirement analysis method to a non-linear fuzzy inference model. Fukuda and Matsuura (1993) also propose to prioritize the customer's requirements by AHP for concurrent design. Although such research work has been developed to elicit customer needs, the work is not so effective, and leads to misread customers' actions and thoughts (Zaltman, 2003).

In summary, most approaches assume product development starts from a clean sheet of paper; however, in practice, most new products evolve from existing products, i.e., so-called variant design (Prebil et al., 1995). Historical data, product evolution paths, and feedback from customers on current products are often considered only implicitly, if not ignored. As a result, product design seldom has the opportunity to take advantage of the wealth of customer requirement information accumulated in existing products. In addition, these methods do not explicitly differentiate the customer preference from the designer's preference of requirement information (Tarasewich and Nair, 2001), nor do any approach exist to handle the mapping from the customer domain to the functional domain effectively. Furthermore, new product development for mass customization is facing the challenge of maintaining the continuity of manufacturing and service operations. Therefore, product definition should effectively preserve the strength of product families to obtain significant cost savings in tooling, learning curves, inventory, maintenance, and so on. This demands a structured approach to product definition and to the capturing of the gestalt of requirement information from previous designs as well as existing product and process platforms.

2.7. Optimal Product Design

While traditional design emphasizes more on the designers' perspective (Tarasewich and Nair, 2001), measuring customer preferences in terms of expected utilities is the primary concern of optimal product design (Krishnan and Ulrich, 2001) or decision-based design (Hazelrigg, 1998). In typical preference-based product design, conjoint analysis (Green and Krieger, 1985) has proven to be an effective means to estimate individual level part-worth utilities associated with individual product attributes. In order to simulate the potential market shares of proposed product concepts, scaled preference evaluations need to be collected from respondents with regard to a subset of multi-attribute product profiles (stimuli) constructed according to a fractional factorial design. From these preference data, idiosyncratic part-worth preference functions are then estimated for each respondent using regression analysis. Attribute level part-worth utilities can also be computed from respondents' simulated choice data, which is called a choice-based conjoint analysis and hence establishes a direct connection between preference and choice (Kuhfeld, 2004). The conjoint-based searching for optimal product designs always results in combinatorial optimization problems because typically discrete attributes are used in conjoint analysis (Kaul and Rao, 1995; Kohli and Sukumar, 1990; Nair et al., 1995).

Multi-attribute utility analysis is widely used to predict composite utilities for any feasible product profile constructed from the underlying attribute level part-worth utilities (Keeney and Raiffa, 1976). It assumes that the utilities of multiple attributes are mutually independent (Wassenaar and Chen, 2001). This may not hold true for a product portfolio where the customer-perceived utility of a particular attribute may change due to the availability of other offerings (for example, comparing with counterpart attributes or levels).

In addition, combining different individual attribute utility functions into a single multi-attribute utility function inevitably involves multi-attribute weighting and normalization. The weights are determined based upon the rank ordering of alternatives; however, a selected alternative may result from the underlying weighting method rather than the quality of the alternative itself (Saari, 2000). Arrow and Raynaud (1986) also point out that group voting always leads to intransitive outcomes, in which the preference of neither a group of decision makers nor a set of criteria can be captured by multi-attribute ranking.

Normalization is often employed to facilitate a comparison of alternatives when attributes involve different dimensions or metrics. It is difficult to judge rigorously a normalizing range, within which each normalized value is determined based on the relative position of the actual attribute level (Wassenaar and Chen, 2001). The weighted sum method is often used to model the relative importance among multiple attributes by assigning different weights to the attributes. The assignment of weights is subjective in nature and often becomes biased when an attribute is correlated to a product’s success (Arrow and Raynaud, 1986). Besides, the weighted sum assumes a linear attribute tradeoff, which is only true for limited variation of attribute levels (Wassenaar and Chen, 2001), but not for the case of a product portfolio, where the number of attributes and their levels may be very large. Hence, Wassenaar and Chen (2001) posit the necessity to use a single criterion approach to decision-based design, which should reflect many different issues regarding customers, design and manufacturing.

2.8. Product Positioning

Product positioning involves decisions about abstract perceptual attributes and customer heterogeneity (Kaul and Rao, 1995). To optimize a new product's positioning, Shocker and Srinivasan (1979) propose a framework using joint space models of customer perceptions and preferences. Joint space analysis entails the mapping between locations of existing products and ideal points for each individual or market segment. The basic principle lies in the multidimensional scaling of customer perceptions via factor analysis, discriminant analysis or similarity scaling (Green and Krieger, 1989). Using a joint mapping of ideal points and product locations, a manager can model customers' choices of existing products, predict their responses to new products, and hence identify optimal new product concepts (Sudharshan et al., 1987).

A number of multidimensional scaling-based algorithms have been developed, dependent upon the number of ideal points (individuals or segments) in the joint space (Kaul and Rao, 1995). Consequently, as the number of ideal points rises, so does the complexity of the optimization problem. Genetic algorithms have been proven to outperform most existing optimal positioning algorithms in dealing with the choice set size heterogeneity between the customer's decision setting and variations in the size of the individuals' choice sets (Balakrishnan and Jacob, 1996).

On the other hand, many algorithms have been formulated with the attempt to improve the realism of the customer choice setting. Deterministic first choice models assume customers choose the offered product that is closest to the ideal point. Probabilistic choice settings postulate the customer's propensity to buy a particular product based on a weighted distance between the ideal point and the offered product. Discrete choice analysis is widely used to identify patterns in choices that customers make among competing products (Ben-Akiva and Lerman, 1985). It allows for the examination of the interaction between market shares and product features, price, service, and promotion with respect to different classes of customers. Sudharshan et al. (1987) find that a probabilistic choice model tends to provide better solutions and larger share projections for new product positioning.

2.9. Product Line Design

Most of the literature on product line design tackles the optimal selection of products by maximizing the surplus – the margin between the customer-perceived utility and the price of the product (Kaul and Rao, 1995; Kohli and Sukumar, 1990). Other objectives widely used in selecting products among a large set of potential products include maximization of profit (Monroe et al., 1976), net present value (Li and Azarm, 2002), a seller's welfare (McBride and Zufryden, 1988), market share (Kohli and Krishnamurti, 1987), and share of choices (Balakrishnan and Jacob, 1996) within a target market. Pullman et al. (2002) combine QFD and conjoint analysis to compare the most preferred features with those profit maximizing features so as to develop designs that optimize product line sales or profit. Kota et al. (2000) propose a product line commonality measure to capture the level of component commonality in a product family. The key issue is to minimize non-value added variations across models within a product family without limiting customer choices.

While numerous papers in the marketing literature deal with the selection problem using various objectives originated from the profit, few of them explicitly model the costs of manufacturing and engineering design (Yano and Dobson, 1998). Dobson and Kalish (1988, 1993) extend the model of Green and Krieger (1985), which does not incorporate prices or costs, to include per-product fixed costs. Recent product line design models allow for more complex cost structures. Raman and Chhajed (1995) and Kim and Chhajed (2001) observe that, in addition to choosing which products to produce, one must also choose the process by which these products are manufactured. Ramdas and Sawhney (2001) consider situations where the fixed cost of a component is shared by two products. Dobson and Yano (1994) allow for complex interactions by admitting per-product fixed costs, resources that can be shared by multiple products, as well as technology choices for each. Morgan et al. (2001) examine the benefits of integrating marketing implications of product mix with more detailed manufacturing cost implications, which sheds light on the impact of alternative manufacturing environment characteristics on the composition of the optimal product line. Chidambaram and Agogino (1999) formulate portfolio analysis as an optimization problem consistent with the manufacturer's goal of incurring minimal costs in the redesign of existing standard components, while meeting customer specifications and satisfying design constraints.

Another dimension in product line design research is price. Robinson (1988) suggests that the most likely competitive reaction to a new product in the short term is a change in price. Choi and DeSarbo (1994) apply game theory to model competing firms' reactions in price and employ a conjoint simulator to evaluate product concepts against competing brands. Dobson and Kalish (1988, 1993) discuss the tradeoffs involved in price setting and choice of the number of products. Guiltinan (1993) emphasizes strategic thinking about the length of a product line by identifying those situations in which variety is an important competitive variable, so as to examine the relationship between variety and cost, to understand the underlying determinants of cannibalization and complementarity, as well as to assess the consequences of not responding to competitive innovations.

Furthermore, product line design basically involves two issues (Li and Azarm, 2002): (1) generation of a set of feasible product alternatives, and (2) subsequent selection of promising products from this reference set to construct a product line. Along this line, existing approaches to product line design can be classified into two categories (Steiner and Hruschka, 2002). One-step approaches aim at constructing product lines directly from part-worth preference and cost/return functions. On the other hand, two-step approaches first reduce the total set of feasible product profiles to a smaller set, and then select promising products from this smaller set to constitute a product line. Most of the literature follows the two-step approach and emphasizes on the maximization of profit contributions in the second step (McBride and Zufryden, 1988; Dobson and Kalish, 1993; Chen and Hausman, 2000). The determination of a product line from a reference set of products is thereby limited to partial models due to the underlying assumption that the reference set is given a priori. Following the two-step approach, Green and Krieger (1985; 1989) introduce several heuristic procedures with consideration of how to generate a reference set appropriately. On the other hand, Kohli and Sukumar (1990) and Nair et al. (1995) adopt the one-step approach, in which product lines are constructed directly from part-worth data rather than by enumerating potential product designs. In general, the one-step approach is more preferable, as the intermediate step of enumerating utilities and profits of a huge number of reference set items can be eliminated (Steiner and Hruschka, 2002). Only when the reference set contains a small number of product profiles can the two-step approach work well. As a result, few papers in the marketing literature allow a large number of attributes for describing a product (Yano and Dobson, 1998).

2.10. Summary

Substantial progress has been achieved in the areas of product family design and platform-based product development. Future research lies in taking a holistic view to find system-wide solutions. More specifically, product family design needs to incorporate more front-end issues such as explicit customer modeling and integration, product demand and market segmentation, along with the economic evaluation of product families.

While the field of product families has matured rapidly over the last decade, there are still a number of relatively unexplored topics that offer numerous opportunities for scholarly inquiry. As discussed in this review, the unanswered questions may be examined through a wide variety of approaches, both theoretically and methodologically. It highlights the motivation to carry out an in-depth study on product portfolio planning, as discussed in the following chapters.

CHAPTER 3
FUNDAMENTALS OF PRODUCT PORTFOLIO PLANNING

This chapter develops a systematic framework for product portfolio planning. The fundamental issues are identified. The technical challenges and key research issues of product portfolio planning are analyzed, and the corresponding solution strategies are proposed.

3.1. Portfolio Strategy

Nowadays, most manufacturing is characterized as mass customization – to satisfy individual customer needs by introducing product proliferation while taking the advantage of mass production efficiency (Pine, 1993). To compete in the marketplace, manufacturers have been seeking for expansion of their product lines and differentiation of their product offerings with the intuitively-appealing belief that large product variety may stimulate sales and thus conduce to revenue (Ho and Tang, 1998). Initially, variety does improve sales as the offerings become more attractive, but as the variety keeps increasing, the law of diminishing returns suggests that the benefits do not keep pace (Child et al., 1991). The consequence of variety explosion manifests itself through several ramifications, including increasing costs due to an exponential growth of complexity, inhibiting benefits from economy of scale, exacerbating inventory imbalances and warehouse suffocation, and jeopardizing the efficiency of manufacturing processes and distribution systems, to name but a few (Wortmann et al., 1997). Facing such a variety dilemma, a company must optimize its external variety with respect to the internal complexity resulting from product differentiation (Tseng and Jiao, 1996).

On the other hand, the practice of making a wide variety of products available and letting customers vote on the shelf seems not only to be wasteful or unaffordable, but also tends to constrain customers' ultimate satisfaction, leading to so-called mass confusion (Huffman and Kahn, 1998). Pine et al. (1993) have reported on the common problem of companies giving customers more choices than they actually want or need. For example, Toyota found that 20% of its product variety accounted for 80% of its sales, and Nissan reportedly offered 87 different types of steering wheels (Chandler and Williams, 1993). Therefore, rather than creating various products in accordance with all anticipating customer needs, it becomes an important campaign for the manufacturer to offer the "right" product variety to the target market.

Such decisions as to the optimal amount of product offerings adhere to the general wisdom as suggested in the Boston Consulting Group's notion of product portfolio strategy (Henderson, 1970). While representing the spectrum of a company's product offerings, the product portfolio must be carefully set up, planned and managed so as to match those customer needs in the target market (Warren, 1983). The customers must be involved; otherwise, it is simply the manufacturer who provides variety for the marketplace (Duray, et al., 2000). The product portfolio strategy has far-reaching impact on the company's business success to achieve financial goals in maximizing return and R&D productivity, to maintain the competitive edge of the business by increasing sales and market share, to allocate scare resources properly and efficiently, to forge the link between project selection and business strategies, to better communicate priorities within the organization both vertically and horizontally, and so on (Cooper et al., 2001). Documented common examples of product portfolios involve film (Jaime, 1998), electronics assembly (Mosher, 1999), photocopiers (Zamirowski, 1999), vehicle options (Roberson, 1999), and commercial aircraft models (Weir, 2000), among many others.

The general gist of planning a product portfolio is exhibited by such procedures as: (1) capture and identify customer real needs; (2) develop conjoint data of market shares; (3) define market segments based on clustering analysis; (4) define portfolio attribute targets using the centroids of clustered results; (5) generate product alternatives in a portfolio by permuting all portfolio attribute levels; and then (6) determine an optimal combination of product alternatives (Wedel and Kamakura, 1998). Therefore, product portfolio planning, in general, involves two main stages: portfolio identification and optimization (Li and Azarm, 2002). The goal of portfolio identification is to capture and understand customer needs effectively and to transform them into specifications of product offerings (e.g., functional features) accordingly. The key issue of portfolio optimization is to determine an optimal setup or configuration of these planned offerings (e.g., the go/no go decision of an offering).

3.2. Product Portfolio Identification

Current researchers and industrial practitioners in this field involve themselves mostly in the economic justification of product portfolio (e.g., product line design), *viz.,* the latter stage of product portfolio planning. They usually imply that the specification of offerings in a product portfolio is given. However, the first issue - how to identify customer needs and generate product portfolio specifications - has received only limited attention. During this phase, many factors need to be considered, including any combination of customer needs, corporate objectives, product ideas and related technological capabilities, etc. Usually, product offerings are represented as a list of functional features and target values. This information is often a mix of quantitative values and qualitative descriptions of product functionality. In most cases, the company may produce a formal document that requires to undergo routinely many amendments along with scrutiny, or to be signed off by many individuals (Prasad, 1996). Even though product portfolio identification is of paramount importance, past research has not addressed it well, nor has actual practice availed to formulate effective means. This may stem from the complications inherent in the product portfolio identification process, as discussed next.

3.2.1. Technical Challenges

To leverage the market benefits of customization and the costs of providing variety, it is reasonable to fulfill mass customization within a company's capabilities in design and production. In practice, this is often achieved by developing product and process platforms (Simpson, 2004; Jiao et al., 2003). A product platform performs as a base product from which product families can variegate designs to satisfy individual customer requirements (Meyer and Lehnerd, 1997). Corresponding to a product platform, production processes can be organized as a process platform in the form of a bill-of-operations (e.g., standard routings), hence facilitating build - or configure-to-order production for given customer orders (Jiao et al., 2000). Both product and process platforms originate from and are thus supposed to conform to a planned product portfolio.

Consistent with the product definition process, product portfolio identification involves a tedious elaboration process enacted among customers, marketing, and designers, as shown in Figure 3-1. Tseng and Jiao (1998) point out the difficulties associated with product definition. Their observations are also supported in the study by Tarasewich and Nair (2001).

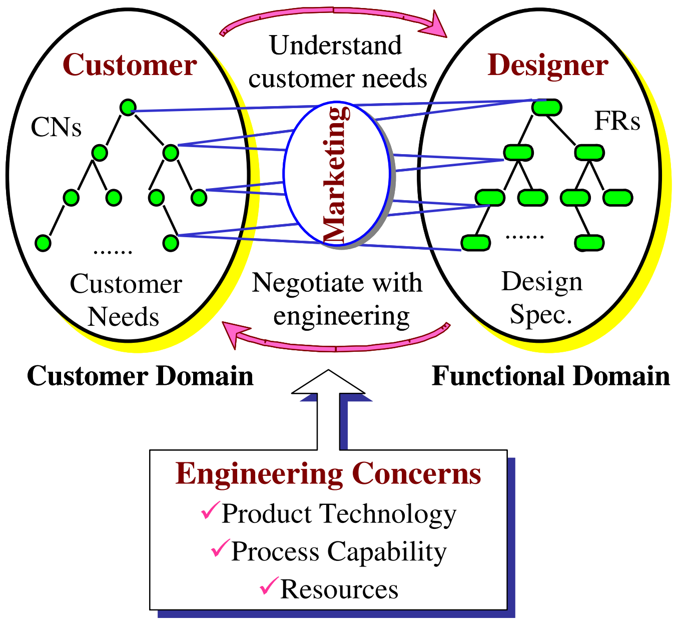


Figure 3-1 Tedious negotiation process inherent in product portfolio identification

First, the customer requirements are normally qualitative and tend to be imprecise and ambiguous due to their linguistic origins. In most cases, requirements are negotiable and conflict with one another, and thus tradeoffs are often necessary. Frequently, customers, marketing staff and designers employ different sets of context to express the requirements. Differences in semantics and terminology always impair the ability to convey requirement information effectively from customers to designers due to their different positions (Zaltman, 2003). The differentiation of requirements in terms of CNs and FRs is of practical significance. An organization should put considerable efforts in capturing the genuine or "real" needs of the customers (CNs), rather than too much focus on the technological issues (FRs) during the early stages of product development (Yan et al., 2002).

Second, there rarely exists any definite structure of requirement information. Variables used to describe requirements are often poorly understood and are usually expressed in abstract, fuzzy, or conceptual terms, leading to work on the basis of vague assumptions and implicit inference. A few researchers have enforced a hierarchical structure or an AND/OR tree structure for the articulation of customer requirements, for example, the requirement taxonomy (Hauge and Stauffer, 1993), the customer attribute hierarchy (Yan et al., 2001), and the FR topology (Tseng and Jiao, 1998). Nevertheless, the non-structured nature of requirement information itself coincides with those findings in nature language processing (Shaw and Gaines, 1996).

Third, the interrelationships (i.e., mapping) between CNs and FRs are often not clearly available in the early stages of design. Customers are often not aware of the underlying coupling and interrelationships among various requirements with regard to product performance. It is difficult, if not impossible, to estimate the consequences (in particular, in terms of economic, scheduling and quality concerns) of specifying different requirements. Christopher et al. (1980) discern customer needs and product specifications and point out the mapping problem between them is the key issue in "design for customers".

Fourth, the specification of requirements results from not only the transformation of customer requirements from those end-users, but also considerations of many engineering concerns, involving any internal customer from downstream of the design team along the product realization process (Du et al., 2003). In practice, product development teams must keep track of a myriad of requirement information derived from different perspectives on the product life-cycle, such as product technologies, manufacturability, reliability, maintainability, and environmental safety, to name but a few (Prudhomme et al., 2003).

Therefore, the process of product portfolio identification can be described as: $\Lambda \leftarrow \Gamma(CNs, Eng.)$, where Λ represents a portfolio of product offerings, CNs indicate the customer needs of end-users, $Eng.$ means engineering considerations associated with CNs, and Γ denotes the mapping relationship from CNs and $Eng.$ to a particular product portfolio, Λ.

3.2.2. Strategy for Solution

Due to the difficulties inherent in the portfolio identification process, reusing knowledge from historical data suggests itself as a natural technique to facilitate the handling of requirement information and tradeoffs among many customer, marketing and engineering concerns. Tseng and Jiao (1998) propose to identify FR patterns from previous product designs for addressing a broad spectrum of domain-specific customer requirements and to organize requirement information during design. In their model, various FRs are grouped according to the similarity among customers (i.e., market segments). The focus is on the functional domain. Du et al. (2003) extend the idea to study the patterns of CNs for better customization and personalization. Chen et al. (2002) apply neural network techniques to construct a customer attribute hierarchy (CAH) in order to improve customer requirement elicitation. Both ideas emphasize on the customer domain. While these proposed solutions emphasize the identification of either CN or FR patterns, the mapping relationship between CNs and FRs has not been taken into account. We assert that FR patterns should not be identified in isolation from those patterns of CNs, and vice versa. The patterns of CN-FR mappings play an important role in bringing engineering concerns into product portfolio identification as well as in determining CN and FR patterns within a cohesive context.

To this end, this research proposes to apply data mining techniques to improve the product portfolio identification process. Data mining has been well recognized for decision support by efficient knowledge discovery of previously unknown and potentially useful patterns of information from past data (Chen et al., 1996). As one of the important applications of data mining, association rule mining lends itself to the discovery of knowledge associated with mappings from CNs to FRs. Based on association rule mining, this research develops an inference system for effective product portfolio identification presented in Chapter 4 in detail.

3.3. Product Portfolio Optimization

Product portfolio optimization has been traditionally dealt with in the management and marketing fields with the focus on portfolio optimization based on customer preferences. The objective is to maximize profit, share of choices, or sales (Urban and Hauser, 1993). Consequently, measuring customer preferences among multi-attribute alternatives has been a primary concern in marketing research. Among many methods developed, conjoint analysis has turned out to be one of the most popular preference-based techniques for identifying and evaluating new product concepts (Green and Krieger, 1985; 1996). A number of conjoint-based models have been developed with particular interests in mathematical programming techniques for optimal product line design (for example, Dobson and Kalish, 1993; Chen and Hausman, 2000). These models seek to determine optimal product concepts using customers' idiosyncratic or segment-level part-worth (i.e., customer-perceived value of a particular level of an attribute) preference functions that are estimated within a conjoint framework (Steiner and Hruschka, 2002). While many methods excel in determining optimal or near-optimal product designs from conjoint data, traditional conjoint analysis is limited to considering input from the customers only, rather than analyzing distinct conjoint data from both customers and engineering concerns (Tarasewich and Nair, 2001).

In the engineering community, product portfolio decisions have been extensively studied with a particular focus on the costs and flexibility issues associated with product variety and mix (for example, MacDuffie et al., 1996; de Groote, 1994; Lancaster, 1990). However the effect of product lines on the profit side of the equation has been seldom considered (Yano and Dobson, 1998). Few industries have developed an effective set of analyses to manage the profit due to variety and the costs due to complexity simultaneously in product portfolio decision making (Otto et al., 2003). It is imperative to take into account the combined effects of multiple product offerings on both profit and engineering costs (Krishnan and Ulrich, 2001). Therefore, product portfolio optimization should be positioned at the crossroads of engineering and marketing, where the interaction between the customer and engineering concerns is the linchpin (Markus and Váncza, 1998). In particular, portfolio decisions with customer-engineering interaction need to address the tradeoffs between economies of scope in profit from the customers and markets and diseconomies of scope in design, production, and distribution at the backend of product fulfillment (Yano and Dobson, 1998). Moreover, achieving a synergy of engineering concerns among products in portfolio planning is deemed to be increasingly beneficial given those efforts in many industries to improve the coordination of design and manufacturing activities across product families and platforms (Morgan et al., 2001; Chidambaram and Agogino, 1999).

3.3.1. Objective Function

Among those customer preference or sellers' value-focused approaches, the objective functions widely used for solving the portfolio optimization problem are typically formulated by measuring the *consumer surplus* – the amount that customers benefit by being able to purchase a product for a price that is less than they would be willing to pay. The idea behind it is that the expected revenue (utility less price) comes from the gain between customer preferences (utility indicating the dollar value that they would be willing to pay) and the actual price they would pay, while the price implies all related costs. A general form is given as follows (see, for example, Green and Krieger, 1985):

$$Maximize \quad \sum_{i=1}^{I}\sum_{j=1}^{J}\left(U_{ij} - p_j\right)P_{ij}Q_i \ , \tag{1}$$

where p_j denotes the price customers actually pay for j-th product; U_{ij} represents the dollar value customers in i-th market segment would willing to pay for j-th product; P_{ij} indicates the probability that customers in i-th segment choose j-th product; and Q_i denotes the market size of i-th market segment.

With more focus on engineering concerns, the optimization problem is approached by measuring the *producer surplus* – the amount that producers benefit by selling at a market price that is higher than they would be willing to sell. The principle is to measure the expected profit (price less cost) based on the margin between the actual price they would receive and the cost (indicating the dollar value they would be willing to sell for), while the price implies customer preference. A general form is given as follows (see, for example, Yano and Dobson, 1998):

$$Maximize \quad \sum_{i=1}^{I}\sum_{j=1}^{J}\left(p_j - C_j^V\right)P_{ij}Q_i - C_j^F \ , \tag{2}$$

where p_j denotes the price the producer would be willing to sell for j-th product; P_{ij} and Q_i bear the same meaning as in Eq.(1); C_j^V and C_j^F indicate the variable cost and allocated fixed cost per product, respectively.

In practice, either the consumer or producer surplus-based optimization approach encounters difficulties when dealing with pricing or cost accounting. As a matter of fact, price competition is one of the most complicated topics in marketing research, where a number of approximations have to be assumed such as price equilibrium, monopolistic producers, oligopoly, market mavenism, etc. (Choi and DeSarbo, 1994). The formidable hindrance of cost estimation lies in its reliance on detailed knowledge of product design and process plans (Jiao and Tseng, 1999b). A complete description of product design, however, is rarely available at the portfolio planning phase, nor exists any well-defined relationship, at the early design stage, between various attribute levels and their cost figures to be committed in manufacturing. More difficult is the allocation of variable and fixed costs among products (Dobson and Kalish, 1993), although a linear-additive fixed cost function is always employed (Moore et al., 1999).

Considering the customer-engineering interaction in product portfolio optimization, the aforementioned economic surpluses should be leveraged from both the customer and engineering perspectives. This research proposes to use a *shared surplus* to leverage both the customer and engineering concerns. Then the objective function can be formulated as follows:

$$Maximize \quad E[V] = \sum_{i=1}^{I} \sum_{j=1}^{J} \frac{U_{ij}}{C_j} P_{ij} Q_i y_j \, , \tag{3}$$

where $E[V]$ denotes the expected value of the shared surplus, V, which is defined as the utility per cost, modified by the probabilistic choice model, P_{ij}, and the market size, Q_i, and C_j indicates the cost of offering specific products, i.e., j-th product. The model development will be clarified in Chapter 5.

The underpinning principle of the shared surplus coincides with the implications of *customer values* in marketing – the customer's expectations of product quality in relation to the actual amount paid for it. It is often expressed as the ratio of the customer-perceived utility to the costs to produce it (Zeithaml, 1988). In addition, introduction of the shared surplus contributes to the maintenance of a consistent measure for the relative comparison of various alternatives on a common ground, while avoiding the intricate pricing and cost estimation problems. This is consistent with the findings reported by Choi and DeSarbo (1994) – "exact cost estimates are not necessary as long as the relative magnitudes are in order." Furthermore, the incorporation of a choice model into customer values enables the modeling of customer decision-making when facing similar product offerings from competitors or even competing products from the same brand. In practice, customer-perceived value of a product tends to decrease if there are counterparts, whereas a premium value can be expected for a unique product owing to limited choices for the customer.

3.3.2. Technical Challenges

In terms of the shared surplus-based optimization model, the main challenges involved in product portfolio optimization are listed next.

First, in most cases, it is hard to measure customer preference (Zaltman, 2003). Customers are always forced to make difficult tradeoffs among competing products. For example, as in real purchase decisions, buyers cannot get all of the best features at the lowest price. It is difficult to simulate the tradeoffs among performance, price, and various product specifications. In addition, customer preferences are heterogeneous. For every two customers whose preferences differ from each other, their appreciations of the same product design may be distinct. The ways they make tradeoffs are also different.

Second, it is difficult to predict the customer choice patterns especially in the marketplace given a competitive situation. The choice patterns vary a lot according to the available product offerings, customer characteristics, etc. In addition, the information that influences the choice patterns is always unobservable.

Third, cost estimation is deemed to be very difficult, especially at the portfolio planning phase. Furthermore, traditional cost accounting by allocating fixed costs and variable costs across multiple products may produce distorted cost-carrying figures due to possible sunk costs associated with investment into product and process platforms. It is quite common in mass customization that design and manufacturing admit resources (and thus the related costs) to be shared among multiple products in a reconfigurable fashion, as well as per-product fixed costs (Moore et al., 1999). In fact, Yano and Dobson (1998) have observed a number of industrial settings, where a wide range of products are produced with very little incremental costs per se, or very high development costs are shared across broad product families, or fixed costs and variable costs change dramatically with product variety. They have pointed out that "the accounting systems, whether traditional or activity-based, do not support the separation of various cost elements".

Fourth, the product portfolio is developed directly from the discrete attributes. As the number of attributes and levels associated with a product increases, so does the number of possible combinations of products for portfolios. A product with nine attributes of three levels each may produce $3^9 = 19683$ possible variants. A product portfolio consisting of maximal three such products may yield $(3^9)^3 + (3^9)^2 + (3^9)^1 = 7.62598 \times 10^{12}$ possible combinations. Complete enumeration to obtain optimal product selections in portfolio optimization becomes numerically prohibitive (Tarasewich and Nair, 2001). The conjoint-based search for an optimal product portfolio always results in a combinatorial optimization problem because typically discrete attributes are used in conjoint analysis (Kaul and Rao, 1995). Nearly all of these problems are known to be mathematically intractable or NP-hard (Nair et al., 1995).

3.3.3. Strategy for Solution

To this end, conjoint analysis, probabilistic choice rules, a pragmatic costing approach, and genetic algorithms are adopted to deal with the technical challenges involved in product portfolio optimization.

(1) Conjoint analysis. Conjoint analysis (CA) has turned out to be one of the most popular preference-based techniques for handling situations in which a decision-maker has to deal with options that simultaneously vary across two or more attributes (Green et al., 2001). Rather than forcing consumers to think separately about individual attributes, conjoint analysis allows the consumers to make judgments about the overall products and then uses statistical analysis to uncover the value system that must be behind the preference judgments.

(2) Probabilistic choice rules. Probabilistic choice rules closely resemble real-world customer choices. The key concept of probabilistic choice rule model is the random utility function (Manski, 1977) where the random utility due to observational deficiencies resulting from the unobserved attributes is addressed.

(3) Pragmatic costing approach. A pragmatic costing approach is developed by Jiao and Tseng (1999b). The idea is to allocate costs to those established time standards from well-practiced work and time studies, thus relieving the tedious tasks for identifying various cost drivers and cost-related activities. The key is to develop mapping relationships from different attribute levels to their expected consumptions of standard times within legacy process capabilities. These part-worth standard time accounting relationships are built into the

product and process platforms (Jiao et al., 2003). Any product configured from available attribute levels is justified based on its expected cycle time.

(4) Combinatorial optimization algorithm. Comparing with traditional calculus-based or approximation optimization techniques, genetic algorithms (GA) have been proven to excel in solving combinatorial optimization problems (Steiner and Hruschka, 2002). The GA approach adopts a probabilistic search technique based on the principle of natural selection by survival of the fittest and merely uses objective function information, and thus is easily adjustable to different objectives with little algorithmic modification (Holland, 1992). An important feature of a GA is that it allows product profiles to be constructed directly from attribute level part-worth data (Kohli and Sukumar, 1990). This is particularly preferable to reference set enumeration if the number of attributes and their levels is large and most multi-attribute products represented by different attribute level combinations are economically and technologically feasible (Nair et al., 1995). Towards this end, this research develops a heuristic GA approach for product portfolio optimization. The details are presented in Chapter 5.

3.4. Summary

As a strategy for portfolio decisions, product portfolio planning involves two stages: portfolio identification and optimization. In this chapter, the implications, technical challenges, and the corresponding solution strategies involved in these two stages are discussed in detail. Product portfolio identification aims at transforming the customer needs into product specifications. The main challenge is the semantic nature of customer needs. The unstructured, ambiguous customer requirement information makes it difficult to identify the real customer needs. Association rule mining is identified as the solution strategy to discover useful patterns associated with requirement analysis enacted among customers, marketing folks, and designers. On the other hand, product portfolio optimization aims at determining an optimal configuration of the identified specifications with the objective of achieving the best shared surplus performance. Genetic algorithms, conjoint analysis, etc., are proposed as the solution strategies to deal with the involved difficulties. The detailed system framework, modeling, and implementation issues for product portfolio identification and optimization are presented in Chapters 4 and 5, respectively.

CHAPTER 4
PRODUCT PORTFOLIO IDENTIFICATION BASED ON ASSOCIATION RULE MINING

This chapter develops explicit decision support to improve product portfolio identification by efficient knowledge discovery methodology. An association rule mining method is proposed to establish the mapping mechanism between customer needs and product specifications. The product portfolio identification problem is formulated in Section 4.1. The methodology and system implementation are proposed for efficient product portfolio identification in Sections 4.2 and 4.3. An application to generate a vibration motor portfolio is presented to validate the feasibility of the proposed methodology and system in Section 4.4. The results of sensitivity analysis evaluate the system performance in Section 4.5. The chapter is concluded with a discussion in Section 4.6.

4.1. Problem Formulation

Figure 4-1 illustrates the principle of product portfolio identification based on association rule mining. In general, customer needs can be described as a set of features or attributes, $A \equiv \{a_1, a_2, \cdots, a_M\}$. Each feature, $a_i \mid \forall i \in [1, \cdots, M]$, may take on one out of a finite set of options, $A_i^* \equiv \{a_{i1}^*, a_{i2}^*, \cdots, a_{in_i}^*\}$. That is, $a_i =:: a_{ij}^* \mid \exists a_{ij}^* \in A_i^*$, where $j = 1, \cdots, n_i$, denotes the j-th option of a_i. Suppose all customers comprise a set, $C \equiv \{c_1, c_2, \cdots, c_S\}$, where S denotes the total number of customers. In the customer domain, requirement information of a particular customer, $c_s \in C \mid \exists s \in [1, \cdots, S]$, can be depicted by a vector of certain options of these features, for example, $\overline{a_s^*} \equiv [a_{13}^*, a_{22}^*, \cdots, a_{M1}^*]$, where a_{13}^* refers to the 3-rd option of feature a_1

as desired by customer c_s, a_{22}^* the 2-nd option of feature a_2, and a_{M1}^* the 1-st option of feature a_M. The population of customers' needs becomes a set, $A^* \equiv \left\{\overline{a_1^*}, \overline{a_2^*}, \cdots, \overline{a_S^*}\right\}$, which characterizes the customer domain.

In the functional domain, the functionality of each product is characterized by a set of FRs, $V \equiv \{v_1, v_2, \cdots, v_N\}$. Each FR, $v_q \mid \forall q \in [1, \cdots, N]$, possesses a few possible values, $V_q^* \equiv \{v_{q1}^*, v_{q2}^*, \cdots, v_{qn_q}^*\}$. That is, v_q =:: $v_{qr}^* \mid \exists v_{qr}^* \in V_q^*$, where $r = 1, \cdots, n_q$, denotes the r-th possible value of v_q. Suppose all existing products comprise a set, $P \equiv \{p_1, p_2, \cdots, p_T\}$, where T refers to the total number of products. The requirement specification of a particular product, $p_t \in P \mid \exists t \in [1, \cdots, T]$, can be represented as a vector of certain FR values of those FRs, for example, $\overline{v_t^*} \equiv [v_{12}^*, v_{21}^*, \cdots, v_{N5}^*]$, where v_{12}^* means product p_t involves the 2-nd value of FR v_1, v_{21}^* the 1-st value of FR v_2, and v_{N5}^* the 5-th value of FR v_N. All the instances of FRs (i.e., FR values) in the functional domain constitute a set, $V^* \equiv \left\{\overline{v_1^*}, \overline{v_2^*}, \cdots, \overline{v_T^*}\right\}$.

Based on the company's sales records and product documentation, we can extract transaction data related to which customer was met with which product. Therefore, transaction data can be summarized as CN-FR pairs in the form of $\left\langle \overline{a_s^*}, \overline{v_t^*} \right\rangle$, where s and t stand for customer ID and product ID, respectively. Each pair of such transaction data not only indicates a specific case of requirement information from both the customer and manufacturer viewpoints, but also implies a particular instance of the mapping relationship between the customer and functional domains.

The difference between the customer and functional domains suggests that what a customer *de facto* perceives is the CNs, rather than FRs. While providing customer-perceived diversity in CNs, the manufacturer must seek for economy of scale in product fulfillment, which is achieved by FRs. In addition, mass customization is by no means to provide whatever customers may want, as excessive variety results in a dramatic increase in costs (Huffman and Kahn, 1998). As postulated in the classic Hotelling-Lancaster model (Hotelling, 1929), some products close together on the spectrum are better substitutes than those further apart. This implies that customers are willing to choose from those products with functional values closest to their desired values if they cannot find any product on the market that exactly matches their desired values. Consumer behavior study also suggests that the consumers falling into the same cluster usually hold the same purchase trend, and thus the customer can be met by providing a product such that the total variations of functionality from what the customer prefers are the smallest. This implies that individual customers within a cluster can most probably be satisfied with a product whose functional values assume the mean values of different expectations by all customers in the same cluster (namely, the centroid of the cluster).

Therefore, in order to take advantage of commonality in product family design, existing instances of FRs, V^*, should be analyzed and clustered according to the similarity among them (Tseng and Jiao, 1996). This process is called FR clustering. The result is a few FR clusters, noted as $X = \{\chi_1, \chi_2, \cdots, \chi_L\}$, where $\chi_l \in X \mid \forall l \in [1, \cdots, L]$, meaning the l-th FR cluster. As a result, all FR instances related to a FR cluster, i.e., $\chi_l \sim V^{*^l} \subset V^*$, can be grouped and represented by the characteristics of χ_l – the mean value of these FR instances, $\mu_l \equiv [x_1^l, x_2^l, \cdots, x_N^l]$, and the variation range of these FR instances within χ_l, $\Delta_l \equiv [\delta_1^l, \delta_2^l, \cdots, \delta_N^l]$. Therefore, each FR cluster can be described as a tuple: $\chi_l = (\mu_l, \Delta_l)$.

Subsequently, these identified FR clusters become the functional specification of product offerings that can be derived from common product platforms and are supposed to be able to accommodate all of the customer needs (Du et al., 2001). In other words, the specification of a product portfolio should cover a group of existing and latent CNs by mapping these needs to the identified FR clusters. At this stage, data mining techniques are applied to figure out the mapping relationship between CNs and FR clusters, noted as $A^* \Rightarrow X$, where an association rule, $\Rightarrow$, indicates an inference from the precedent (A^*) to the consequence (X). As a result, a product portfolio specification, Λ, consists of two elements: FR clusters and mappings from CNs to FR clusters, namely, $\Lambda = \langle X, \Rightarrow \rangle$.

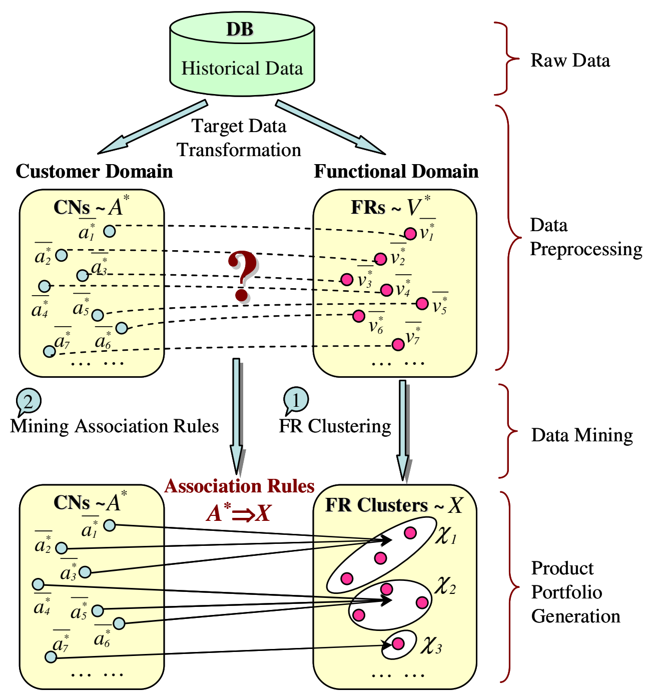


Figure 4-1 Product portfolio identification based on association rule mining

4.2. Methodology

4.2.1. FR Clustering

Clustering analysis refers to a process of grouping a set of physical or abstract objects into classes of similar objects. A cluster is a collection of objects that are similar to one another within the same cluster yet dissimilar to the objects in other clusters (Han and Kamber, 2001). The specification of FRs usually presents in the form of numerical, binary or nominal variables. To handle both quantitative and qualitative variables, this research adopts a fuzzy clustering approach to FR clustering. Fuzzy equivalence relations excel in revealing the similarity between any two objects involving subjectiveness and imprecision (Zimmermann, 1985). Fuzzy clustering is used to create a hierarchical decomposition of the given set of objects, in which each object forms a separate group and successively the objects or groups close to one another are merged at different similarity levels. In this case, historical data about FR instances contained in the platform can be used to measure the similarity degree based on the compatibility of FR value ranges. Comparing with the most popular clustering technique, k-means method, fuzzy clustering partitions FR instances based on the similarity degree that is derived from the real data of FR values, rather than subjectively pre-defined clusters. By varying the similarity threshold, different clusters can be derived to justify the granularity criteria for the product portfolio.

Given a collection of objects (i.e., FR instances), $Z = V^* = \{\overline{v_t^*} \mid \forall t = 1, \cdots, T\}$, a fuzzy set F in Z is defined as a set of ordered pairs: $F = \{(z, \varphi_F(z)) \mid z \in Z\}$, where $\varphi_F(z)$ is called the membership function of z in F that maps Z to $[0,1]$. The membership function is also referred to as the degree of compatibility or degree of truth. A certain set of objects that belong to the fuzzy set F at least to the degree λ is called the λ-cut.

Assume Z is a finite, non-empty set called the universe. Let R be a fuzzy relation in $Z \times Z$, that is, $R = \{(x, y) \mid \forall (x, y) \in Z \times Z)\}$, then according to (Lin and Lee, 1996):

(1) R is reflexive if $\varphi_R(z,z) = 1 \mid \forall z \in Z$;

(2) R is symmetric if $\varphi_R(z,x) = \varphi_R(x,z) \mid \forall x,z \in Z$; and

(3) R is max-min-transitive if $\varphi_R(z,x) \geq max_{y \in Y}\{min\{\varphi_R(z,y),\varphi_R(y,x)\}\}$, i.e., $R \circ R \subseteq R$.

If R is reflexive and symmetric, R is said to be a fuzzy compatible relation. If R is reflexive, symmetric, and transitive, R is said to be a fuzzy equivalence relation. Fuzzy clustering becomes a set of T objects of Z to be clustered, given a fuzzy compatible relation R defined on Z. Assume R^t denotes the *t*-th power of fuzzy relation R, i.e., $R^t = R^{t-1} \circ R$, where $\circ$ is max-min composition. Then the max-min-transitive closure of R, denoted as R^*, can be defined as $R^* = \bigcup_{i=1}^{T} R^i$. Therefore, R^* is a fuzzy equivalence relation. Assume $0 \leq \lambda \leq 1$ and let $R^*_\lambda = \{(z,x) \mid \varphi_{R^*}(z,x) \geq \lambda, \forall x,z \in Z\}$. Then we know from (Wang and McCauley-Bell, 1996) that:

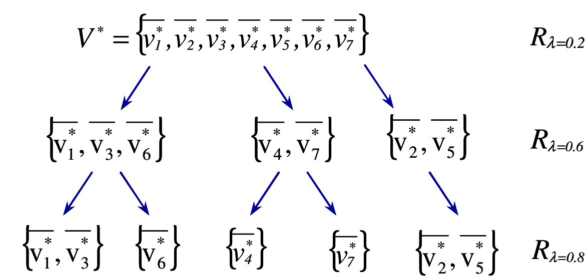
(1) R^*_λ is an equivalence relation on Z; and

(2) Let $G_{R^*_\lambda}$ denote the partition on Z induced according to R^*_λ. Then for each $B \in G_{R^*_\lambda}$, there exists $E \in G_{R^*_{\lambda'}}$, so that $B \subseteq E$, as long as $\lambda' \leq \lambda$.

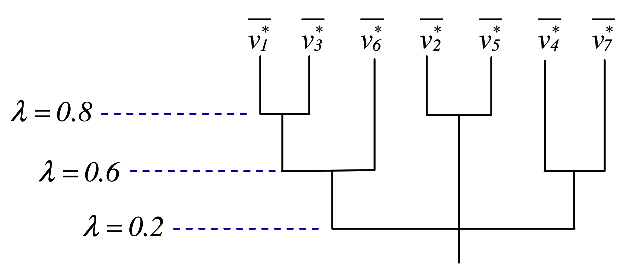
As a result, λ-cut of fuzzy equivalence relation R^*, R^*_λ, becomes an equivalence relation. As λ increases, a finer partition can be achieved. With a hierarchy of partitions of objects, k-clusters of objects can be identified. Figure 4-2 illustrates the nested partitions corresponding to a fuzzy equivalence relation defined based on the FR instances. Given different values of the similarity threshold, λ, different clustering results can be obtained.

$$
R = \begin{array}{c|ccccccc}
V^* & \overline{v_1^*} & \overline{v_2^*} & \overline{v_3^*} & \overline{v_4^*} & \overline{v_5^*} & \overline{v_6^*} & \overline{v_7^*} \\
\hline
\overline{v_1^*} & 1 & 0.2 & 1 & 0.4 & 0.3 & 0.7 & 0.5 \\
\overline{v_2^*} & 0.2 & 1 & 0.2 & 0.3 & 0.8 & 0.3 & 0.4 \\
\overline{v_3^*} & 1 & 0.2 & 1 & 0.4 & 0.1 & 0.6 & 0.3 \\
\overline{v_4^*} & 0.4 & 0.3 & 0.4 & 1 & 0.1 & 0.5 & 0.7 \\
\overline{v_5^*} & 0.3 & 0.8 & 0.1 & 0.1 & 1 & 0.2 & 0.5 \\
\overline{v_6^*} & 0.7 & 0.3 & 0.6 & 0.5 & 0.2 & 1 & 0.3 \\
\overline{v_7^*} & 0.5 & 0.4 & 0.3 & 0.7 & 0.5 & 0.3 & 1
\end{array}
$$

(a) A fuzzy equivalence relation defined on V^*



(b) Nested partitions of V^* induced according to R_λ



(c) Different FR clusters resulted from different values of similarity threshold

Figure 4-2 Fuzzy clustering of FR instances

4.2.2. Association Rule Mining

FR clustering can separate data items into clusters of items, but it cannot explain the clustering results specifically. It needs other methods to figure out the underlying mechanisms of CN-FR mapping between the customer and functional domains. Knowledge is usually represented in the form of rules. Rules are used for deducing the degree of association among variables, mapping data into predefined classes, identifying a finite set of categories or clusters to describe the data, etc. Therefore, this research employs association rules to explain the meaning of each FR cluster as well as the mapping of CNs to each cluster. Association rule mining is one of the major forms of data mining and is perhaps the most common form of knowledge discovery in unsupervised learning systems (Chen et al., 1996). Association rules are produced by finding the interesting associations or correlation relationships among a large set of data items. The flexibility of association rule induction lies in its capability to deal with those qualitative data that cannot be treated by traditional operations research methods.

The basic problem of mining association rules is introduced by Agrawal et al. (1993). Let $I = \{i_1, i_2, \cdots, i_m\}$ be a set of literals, called items. Let DB be a database of transactions, where each transaction, T, is a set of items such that $T \subseteq I$, and each transaction is associated with an identifier, called TID. Given $Z \subseteq I$, a transaction T contains Z if and only if $Z \subseteq T$. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subseteq I$, $Y \subseteq I$, and $X \cap Y = \varnothing$. The association rule $X \Rightarrow Y$ holds in DB with confidence c if $c\%$ of the transactions in DB that contain X also contain Y. This is taken to be a conditional probability, $P(y/x \mid \forall x \in X, \forall y \in Y)$. The association rule $X \Rightarrow Y$ has support s in DB if $s\%$ of the transactions in DB contain X and Y. The support is taken to be a probability, $P(x \wedge y \mid \forall x \in X, \forall y \in Y)$.

While the confidence denotes the strength of implication, the support indicates the frequencies of the occurring patterns in the rule. Given a minimum confidence threshold, min_conf, and a minimum support threshold, min_sup, the problem of mining association rules becomes a search for all the association rules whose confidence and support are larger than the respective thresholds. Based on whether or not they can meet the thresholds (min_conf and min_sup), association rules are distinguished between strong rules and weak ones. A set of items is referred to as an itemset. An itemset that contains k items is called a k-itemset. Given a minimum support threshold, min_sup, an itemset is called large if its support is no less than min_sup. Association rule mining involves a two-step process (Agrawal et al., 1993):

(1) Discover all large itemsets whose support is larger than the predetermined minimum support threshold. Itemsets with minimum support are called frequent itemsets; and

(2) Generate strong association rules from the large itemsets.

The most crucial factor affecting the performance of mining association rules lies in the first step. After the large itemsets are identified, the corresponding association rules can be derived in a straightforward manner. Efficient counting of large itemsets is hence the focus of most prior studies on algorithms for mining association rules.

4.3. ARMS Architecture and Implementation

Knowledge discovery for CN-FR mapping mechanisms is an interactive and iterative process. Based on association rule mining, an inference system can be constructed for effective product portfolio identification. Figure 4-3 illustrates the architecture of such an association rule mining system (ARMS). The system involves four consecutive stages interacted one and another to achieve the goals, namely the data preprocessing, FR clustering, association rule mining and rule evaluation, and presentation modules. First, historical data are selected and transformed to proper target data sets, which are further analyzed and preprocessed for subsequent mining procedures. The data mining procedure then starts to search for interesting patterns using the clustering module and rule mining module. After mining of association rules, rule evaluation is performed to eliminate any weak rules under the initial criteria predefined by the system. The useful rules are stored with different presentation styles in the knowledge base that may be in the forms of case bases, rule bases, and others.

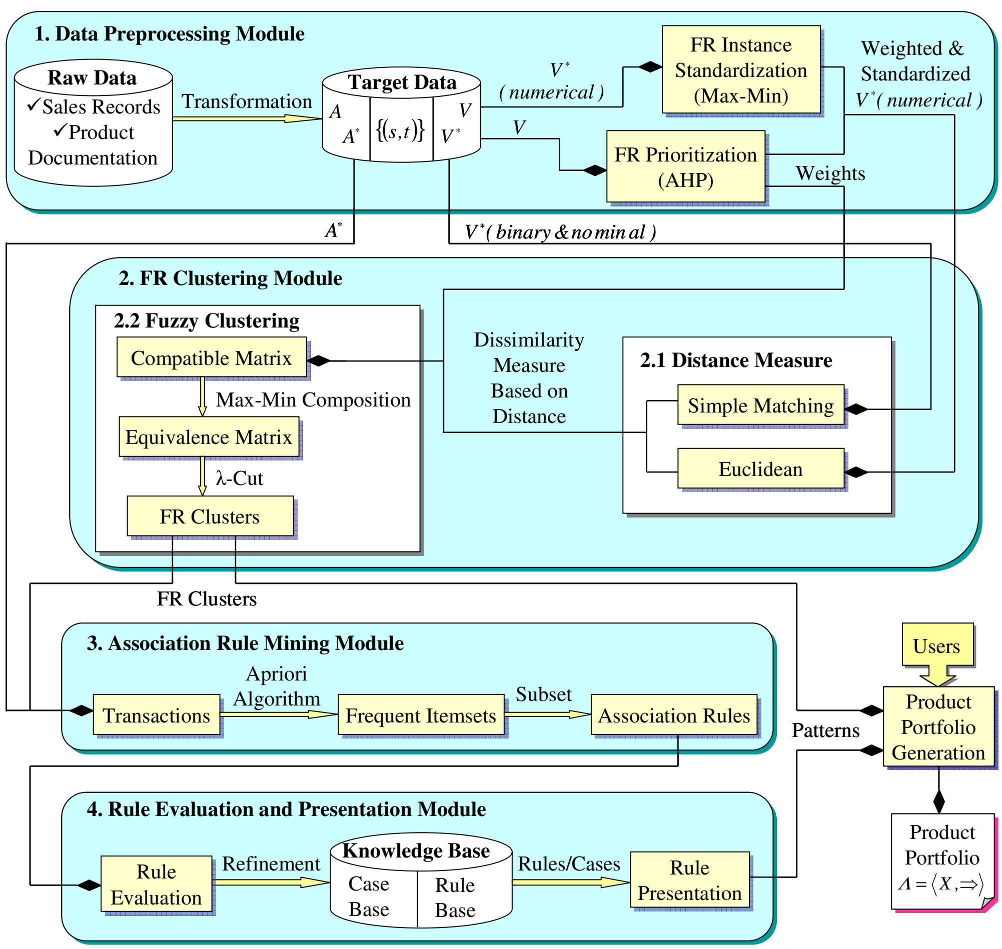


Figure 4-3 ARMS system architecture

4.3.1. Data Preprocessing

Before proceeding to rule mining of data sets, raw data must be preprocessed in order to be useful for knowledge discovery. Three tasks are involved at this stage, as described next.

(1) Target data transformation. Generally, there are lots of data records in a company's databases. Only those records that correlate closely with the mining purpose are taken into account. Based on raw data stored in the company, target data sets should be identified, regarding such data cleaning and filtering tasks as integration of multiple databases, removal of noises, handling of missing data files, etc.

All target data should be organized into a proper transaction database. This involves understanding the variables, selection of attributes and metrics, and identification of entity relationships among data. Within the ARMS, sales records and product documentation are transformed into transaction data (TID). Transaction data consists of customer records (C) and their ordered products (P). Each customer is described by his/her choices of certain options (A^*) for some functional features (A). The product ordered by this customer is described by specific values (V^*) of related FRs (V). The results of CN-FR mappings, i.e., $\left\langle \overline{a_s^*}, \overline{v_t^*} \right\rangle$, are embodied in the transaction records ($\left\langle C, P \right\rangle$). Figure 4-4 shows the entity relationships among these target data sets.

(2) Prioritization of FR variables. The specification of FRs involves multiple variables, i.e., $V = \{v_q\} | \forall q = 1, \cdots, N$. These FR variables contribute to the overall functionality of a product differently – some may play more roles than others. Hence, FR variables should be prioritized to differentiate their different effects, in particular those important ones. The relative importance of FR variables is usually quantified by assigning different weights. That

is, each v_q is associated with a weight, w_q, subjective to $\sum_{q=1}^{N} w_q = 1$. For the ARMS, the AHP (Saaty, 1980) is adopted for the prioritization of FR variables, owing to its advantages in maintaining consistence among a large number of variables through pair-wise comparisons.

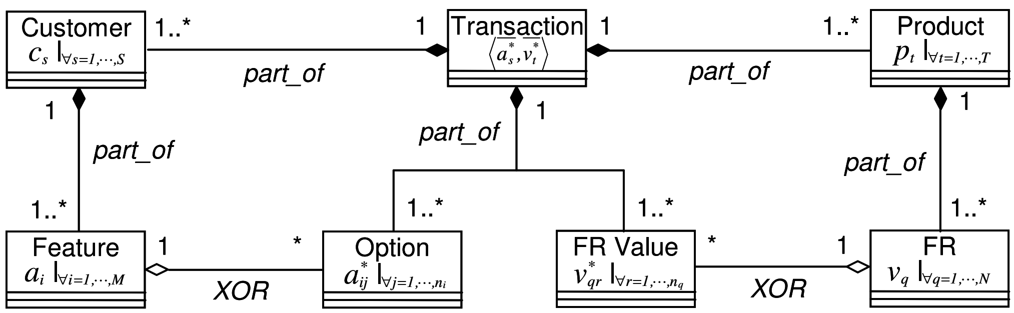


Figure 4-4 Entity relationships of target data sets

(3) Standardization of FR values. Prior to clustering analysis of FR instances, all V^* data needs to be transformed into standard forms because FR variables may involve different metrics and ranges of values. In general, expressing a variable in smaller units will lead to a larger range for that variable, and thus a larger impact on the clustering structure. To avoid dependence on the choice of different metrics or dominance of certain variables over others, those FR instances that are of a numerical type should be standardized to become dimensionless. This is achieved by normalization. Many methods are available such as the z-score method, the max-min normalization method (Han and Kamber, 2001). The ARMS adopts the latter method. Assume some of the FR variables, $v_k \in V \mid \forall k = 1,\cdots,Q \le N$, are of numerical type. It means that their values, $v_{kr}^* \in V_k^* \mid \forall r = 1,\cdots,n_k$, are numerical, where n_k refers to the number of values that v_k can assume. Applying the max-min method, each individual value of v_k, v_{kr}^*, can be normalized to become a dimensionless number ranged between 0 and 1, that is,

$$N\_v_{kr}^{*} = \frac{v_{kr}^{*} - min\left\{v_{kj}^{*} \mid \forall j = 1,\cdots,n_k\right\}}{max\left\{v_{kj}^{*} \mid \forall j = 1,\cdots,n_k\right\} - min\left\{v_{kj}^{*} \mid \forall j = 1,\cdots,n_k\right\}}, \tag{4}$$

where $N\_v_{kr}^{*}$ denotes the normalized value for the r-th value of FR v_k, v_{kr}^{*} is the original values of v_k, and $max\{v_{kj}^{*} \mid \forall j = 1,\cdots,n_k\}$ and $min\{v_{kj}^{*} \mid \forall j = 1,\cdots,n_k\}$ are the maximum and minimum values among all values of v_k with size-n_k, respectively.

In some cases, those non-numerical FR instances, such as nominal FRs, should be transformed into normalized numerical values. For instance, the data type of FR "coating material" is originally a nominal type (i.e., character strings). A scaling transformation can be applied such that, for example, "Au coating" is supplanted by 0.2, "Alloy coating" becomes 0.4, and so on. When all FR instances possess the same measurements and ranges, we can proceed to the FR clustering process.

4.3.2. FR Clustering

Within the ARMS, FR clustering includes two steps: distance measure and fuzzy clustering. As a preparatory stage for fuzzy clustering, the distance measure module measures the dissimilarity between FR instances in order to define the fuzzy compatible relations among such data objects.

(1) Distance measure. In general, each FR instance, $\overline{v_t^{*}} = [v_{1t}^{*}, v_{2t}^{*}, \cdots, v_{qt}^{*}, \cdots, v_{Nt}^{*}] \in V^{*}$, where $\forall v_{qt}^{*} \equiv v_{qr}^{*}$, $\exists v_{qr}^{*} \in V_q^{*}$, $\forall r = 1,\cdots,n_q$, may involve three types of FR variables: numerical, binary, and nominal. For example, v_{1t}^{*} may be a numerical value while v_{2t}^{*} may be a binary or nominal value. The distance between any two FR instances indicates their dissimilarity and thus is measured as a composite distance of three distance components corresponding to these three types of FR variables.

Numerical FRs — A number of methods for distance measure have been proposed for purpose of numerical clustering, including the Euclidean distance, Manhattan distance, Minkowski distance and weighted Euclidean distance measure (Han and Kamber, 2001). The ARMS employs the weighted Euclidean distance. It is the most popular method for calculating the distance between multi-dimensional objects, while still considering the relative importance of each dimension. It is computed as the following,

$$d_{numerical}\left(\overline{v_i^*}, \overline{v_j^*}\right) = \sqrt{\sum_{q=1}^{Q}\left(w_q\left(N\_v_{qi}^* - N\_v_{qj}^*\right)\right)^2}\,, \tag{5}$$

where $d_{numerical}\left(\overline{v_i^*}, \overline{v_j^*}\right)$ indicates the numerical distance between two FR instances, $\overline{v_i^*}$ and $\overline{v_j^*}$, $\forall \overline{v_i^*}, \overline{v_j^*} \in V^*$, w_q means the relative importance of the q-th numerical FR variable, $v_q \in V^{numerical} \subseteq V$, Q represents the total number of numerical FR variables among the total size-N FR variables ($Q \leq N$), and $N\_v_{qi}^*$ and $N\_v_{qj}^*$ denote the normalized values of original v_{qi}^* and v_{qj}^* according to Eq.(4), respectively,

Binary FRs — A binary variable assumes only two states: 0 or 1, where 0 means the variable is absent, and 1 means it is present. The ARMS uses a well-accepted coefficient for assessing the distance between symmetric binary variables, called the simple matching coefficient (Han and Kamber, 2001). It is calculated as the following,

$$d_{binary}\left(\overline{v_i^*}, \overline{v_j^*}\right) = \frac{\alpha_2 + \alpha_3}{\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4}\,, \tag{6}$$

where $d_{binary}\left(\overline{v_i^*}, \overline{v_j^*}\right)$ indicates the binary distance between two FR instances, $\overline{v_i^*}$ and $\overline{v_j^*}$, $\forall \overline{v_i^*}, \overline{v_j^*} \in V^*$, α_1 is the total number of binary FR variables in V (i.e., $v_q \in V^{binary} \subseteq V$) that equal to 1 for both $\overline{v_i^*}$ and $\overline{v_j^*}$, α_2 is the total number of binary FR variables that equal to 1 for $\overline{v_i^*}$ but 0 for $\overline{v_j^*}$, α_3 is the total number of binary FR variables that equal to 0 for $\overline{v_i^*}$ but 1 for $\overline{v_j^*}$, and α_4 is the total number of binary FR variables that equal to 0 for both $\overline{v_i^*}$ and $\overline{v_j^*}$.

Nominal FRs — A nominal variable can be regarded as a generalization of a binary variable in that it can take on more than two states. This type of variable can not be expressed by numerical values but by qualitative expressions with more than one option. Therefore, the simple matching coefficient can also be used here to measure the nominal distance between two FR instances containing nominal FR variables (Han and Kamber, 2001):

$$d_{nominal}\left(\overline{v_i^*},\overline{v_j^*}\right)=\frac{\beta-\gamma}{\beta}, \tag{7}$$

where $d_{nomical}\left(\overline{v_i^*},\overline{v_j^*}\right)$ indicates the nominal distance between two FR instances, $\overline{v_i^*}$ and $\overline{v_j^*}$, $\forall\overline{v_i^*},\overline{v_j^*}\in V^*$, γ means the total number of nominal FR variables in V (i.e., $v_q \in V^{nomical} \subseteq V$) that assume the same states for $\overline{v_i^*}$ and $\overline{v_j^*}$; and β is the total number of nominal variables among total size-N FR variables ($\beta \le N$).

Given a set of FR variables, $V \equiv \{v_1, v_2, \cdots, v_N\}$, every FR instance assumes a certain value for each of the FR variable, and thus consists of a combination of numerical, binary and/or nominal FR values, that is, $V^{numerical} \cup V^{binary} \cup V^{nominal} = V$. As a result, the overall distance between $\overline{v_i^*}$ and $\overline{v_j^*}$ comprises three components: the numerical, binary and nominal distances. A composite distance can thus be obtained by the weighted sum:

$$d\left(\overline{v_i^*},\overline{v_j^*}\right)=W_{numerical}d_{numerical}\left(\overline{v_i^*},\overline{v_j^*}\right)+W_{binary}d_{binary}\left(\overline{v_i^*},\overline{v_j^*}\right)+W_{nominal}d_{nominal}\left(\overline{v_i^*},\overline{v_j^*}\right), \tag{8}$$

$$\sum\left(W_{numerical}+W_{binary}+W_{nominal}\right)=1, \tag{9}$$

where $W_{numerical}$, W_{binary} and $W_{nominal}$ refer to the relative importance of numerical, binary and nominal distances, respectively. These weights can be determined in the similar way as that of FR variables – applying the AHP.

(2) Fuzzy clustering. The first step of fuzzy clustering is to define a fuzzy compatible relation, R, for a given set of FR instances, $V^* = \{\overline{v_1^*}, \overline{v_2^*}, \cdots, \overline{v_T^*}\}$. The R is constructed in a matrix form, that is, $R = \left[\rho\left(\overline{v_i^*}, \overline{v_j^*}\right)\right]_{T\times T} \mid \forall\left(\overline{v_i^*}, \overline{v_j^*}\right) \in V^* \times V^*$, where $\left(\overline{v_i^*}, \overline{v_j^*}\right)$ suggests pair-wise relationships among FR instances. Within the context of FR clustering, R is called the compatible matrix. A matrix element $\rho\left(\overline{v_i^*}, \overline{v_j^*}\right)$ indicates the similarity grade between any two FR instances, $\overline{v_i^*}$ and $\overline{v_j^*}$. As a measure of similarity, it can be derived from the aforementioned dissimilarity measure that is determined by the distance between FR instances. Then we have the following:

(a) Normalize the distance measure between $\overline{v_i^*}$ and $\overline{v_j^*}$ based on Eqs. (4) and (8), i.e.,

$$N\_d\left(\overline{v_i^*}, \overline{v_j^*}\right) = \frac{d\left(\overline{v_i^*}, \overline{v_j^*}\right) - min\left\{d\left(\overline{v_x^*}, \overline{v_y^*}\right) \mid \forall x, y = 1, \cdots, T\right\}}{max\left\{d\left(\overline{v_x^*}, \overline{v_y^*}\right) \mid \forall x, y = 1, \cdots, T\right\} - min\left\{d\left(\overline{v_x^*}, \overline{v_y^*}\right) \mid \forall x, y = 1, \cdots, T\right\}}, \qquad (10)$$

where $N\_d\left(\overline{v_i^*}, \overline{v_j^*}\right) \in [0,1]$ is the normalized value of original distance $d\left(\overline{v_i^*}, \overline{v_j^*}\right)$, and $d\left(\overline{v_x^*}, \overline{v_y^*}\right) \mid \forall \overline{v_x^*}, \overline{v_y^*} \in V^*$ stands for a distance measure between any two FR instance based on pair-wise comparisons, $(x, y) \in T \times T$; and

(b) Derive the similarity grade $\rho\left(\overline{v_i^*}, \overline{v_j^*}\right)$ from the normalized distance measure $N\_d\left(\overline{v_i^*}, \overline{v_j^*}\right)$, since it indicates the dissimilarity, i.e.,

$$\rho\left(\overline{v_i^*}, \overline{v_j^*}\right) = 1 - N\_d\left(\overline{v_i^*}, \overline{v_j^*}\right). \tag{11}$$

Hence, we have $0 \le \rho\left(\overline{v_i^*}, \overline{v_j^*}\right) \le 1$. In addition, we can infer that $\rho\left(\overline{v_i^*}, \overline{v_i^*}\right) = 1 \mid \forall i = 1, \cdots, T$, suggesting that R is reflexive, and $\rho\left(\overline{v_i^*}, \overline{v_j^*}\right) = \rho\left(\overline{v_j^*}, \overline{v_i^*}\right) \mid \forall i, j = 1, \cdots, T$, suggesting R is symmetrical. As a result, matrix $R = \left[\rho\left(\overline{v_i^*}, \overline{v_j^*}\right)\right]_{T \times T} \mid \rho\left(\overline{v_i^*}, \overline{v_j^*}\right) \in [0,1]$ becomes a fuzzy compatible relation defined on V^*. Representing a subset of Cartesian product $V^* \times V^*$, matrix R is called a fuzzy compatible matrix.

The second step is to construct a fuzzy equivalence relation for V^* with transitive closure of the fuzzy compatible relation defined above. The fuzzy compatible matrix R is a fuzzy equivalence matrix if and only if the transitive condition can be met, i.e.,

$$\rho\left(\overline{v_i^*}, \overline{v_j^*}\right) \ge max\left\{min\left\{\rho\left(\overline{v_i^*}, \overline{v_z^*}\right), \rho\left(\overline{v_z^*}, \overline{v_j^*}\right) \mid \forall \overline{v_i^*}, \overline{v_z^*}, \overline{v_j^*} \in V^*\right\}\right\}. \tag{12}$$

To convert a compatible matrix to an equivalence matrix, the "continuous multiplication" method is often used. Multiplication in fuzzy relations is also known as max-min composition (Lin and Lee, 1996). Let $R\left(\overline{v_i^*}, \overline{v_z^*}\right)$ and $R\left(\overline{v_z^*}, \overline{v_j^*}\right)$ be two fuzzy compatible relations, then $R \circ R = \left[\left(\overline{v_i^*}, \overline{v_j^*}\right), max\left\{min\left\{\rho\left(\overline{v_i^*}, \overline{v_z^*}\right), \rho\left(\overline{v_z^*}, \overline{v_j^*}\right)\right\}\right\}\right]$ is also a fuzzy compatible relation. To achieve the max-min-transitive closure of R, the flowchart of max-min composition is shown in Figure 4-5.

The third step is to determine λ-cut of the equivalence matrix. The λ-cut is a crisp set, R_λ, that contains all the elements of the universe, V^*, such that the similarity grade of R is no less than λ. That is,

$$R_\lambda = \left[\tau\left(\overline{v_i^*}, \overline{v_j^*}\right)\right]_{T \times T}, \tag{13}$$

where $\tau\left(\overline{v_i^*},\overline{v_j^*}\right)=\begin{cases}1 & if\ \rho\left(\overline{v_i^*},\overline{v_j^*}\right)\geq\lambda \\ 0 & if\ \rho\left(\overline{v_i^*},\overline{v_j^*}\right)<\lambda\end{cases},\quad \rho\left(\overline{v_i^*},\overline{v_j^*}\right)\in[0,1].$ (14)

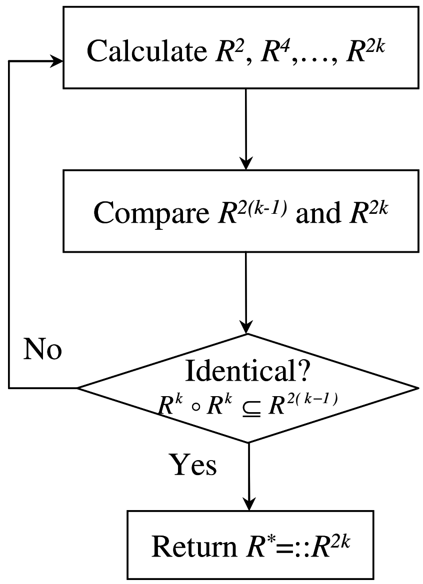


Figure 4-5 The flowchart of converting a compatible matrix to an equivalent matrix

Then each λ-cut, R_λ, is an equivalence relation representing the presence of similarity among FR instances to the degree λ. For this equivalence matrix, there exists a partition on V^*, $\psi(R_\lambda)$, such that each compatible matrix is associated with a set, $\psi(R)=\{\psi(R_\lambda)\}$. The ARMS applies a netting method (Yang and Gao, 1996) to identify partitions of FR instances with respect to a given equivalence matrix. The netting method is a technique dealing with the equivalent matrix. It works via a threshold to indicate the similarity degree between objects, and thus partitioning similar objects into the same cluster. By varying the threshold, different clusters can be derived. The procedure of generating a fuzzy netting graph is summarized as the following,

(a) Fill the signals of the elements in the diagonal;

(b) Replace element 1 as signal * and element 0 as blank;

(c) Connect longitude and latitude to the nodes where the signals * are located; and

(d) Assign the elements that are connected through the nodes into the same cluster.

The value of $\lambda \in [0,1]$ indicates the similarity threshold of a λ-cut. Given an equivalence matrix, different clustering results can be obtained according to individual similarity thresholds, as shown in Figure 4-2(c). In practice, the value of λ is often determined by domain experts with many practical considerations (Lin and Lee, 1996). Furthermore, latent and future customer needs, trends of product and process technologies, repeatability in design and manufacturing, ease of configuration, core competencies, and many others, are also important dimensions of decision making for the threshold.

Finally, with the hierarchy of partitions of objects, k-clusters of objects can be identified. Each FR cluster, $\chi_l = (\mu_l, \Delta_l) \mid \forall l = 1,\cdots,L$, is described by a vector of its mean, $\mu_l = [x_q^l]_N$, and a vector of its variation range, $\Delta_l = [\delta_q^l]_N$.

For a numerical FR value (i.e., $v_{qt}^* \sim v_q \in V^{numerical}$), the mean value and the variation range are calculated as the following,

$$x_q^l = \sum_{t=1}^{n_l} v_{qt}^* \Big/ n_l \,, \tag{15}$$

$$\delta_q^l = max\left\{\left|v_{qt}^* - x_q^l\right| \mid \forall t = 1,\cdots,n_l\right\}, \tag{16}$$

where $\forall q \in [1,\cdots,N]$, $\forall \overline{v_t^*} = [v_{qt}^*]_N \in V^*$, and n_l refers to the number of FR instances associated with the l-th FR cluster, i.e., $\forall \overline{v_t^*} \sim \chi_l \mid \forall t = 1,\cdots,n_l \le T$.

For a binary FR value (i.e., $v_{qt}^* \sim v_q \in V^{binary}$), the mean value and the variation range are determined as the following,

$$x_q^l = \begin{cases} 1 & if\ \alpha_Y \geq \alpha_N \\ 0 & if\ \alpha_Y < \alpha_N \end{cases}, \tag{17}$$

$$\delta_q^l = 0, \tag{18}$$

where $\forall q \in [1, \cdots, N]$, $\forall \overline{v_t^*} = [v_{qt}^*]_N \in V^*$, $\alpha_Y + \alpha_N = n_l$, n_l refers to the number of FR instances associated with the l-th FR cluster, i.e., $\forall \overline{v_t^*} \sim \chi_l \mid \forall t = 1, \cdots, n_l \leq T$, α_Y is the total number of FR instances that assume a 1-state for v_q, and α_N is the total number of FR instances that assume a 0-state for v_q.

For a nominal FR value (i.e., $v_{qt}^* \sim v_q \in V^{nominal}$), the mean value and the variation range are determined as the following,

$$x_q^l = v_{qr}^* \mid r = max(\alpha_r), \tag{19}$$

$$\delta_q^l = 0, \tag{20}$$

where $\forall q \in [1, \cdots, N]$, $\forall \overline{v_t^*} = [v_{qt}^*]_N \in V^*$, v_{qr}^* represents the r-th state of v_q that possesses n_q possible states, i.e., $\exists r \in [1, n_q]$, and α_r is the total number of FR instances that assume a v_{qr}^*-state for v_q.

4.3.3. Association Rule Mining

As reviewed in Section 4.1, traditional association rule mining ($Z \Rightarrow Y$) conforms to the general model of market basket analysis, where all items are assumed to belong to one itemset of transaction data ($Z \subseteq I$ and $Y \subseteq I$). In the ARMS scenario, rule mining involves two different itemsets, that is, $Z \subseteq A^*$ and $Y \subseteq V^*$, corresponding to the customer and functional domains, respectively. Based on the clustered FR instances, association rules regarding the mappings between individuals A^* and V^* turn out to be the association rules mapping A^* to FR clusters, X, that is, $A^* \Rightarrow X$. Therefore, the ARMS's transaction data comprises these two itemsets, i.e., $DB \sim \langle A^*, X \rangle$, where $A^* = \{\overline{a_s^*} \mid \forall s = 1, \cdots, S\}$ and $X = \{\chi_l \mid \forall l = 1, \cdots, L\}$. Itemset A^* consists of a number of sales records of CNs embodied in various combinations of customer choices for diverse options of features, i.e., $\{a_{ij}^* \mid \forall i = 1, \cdots, M, \forall j = 1, \cdots, n_i\}$, where a_{ij}^* corresponds to the *j*-th option of feature a_i, which possesses n_i possible options. Each customer's order indicates a particular combination of these options, i.e., $\overline{a_s^*} = [a_{ij}^*]_M$. Itemset X comprises a set of FR clusters in the form of mean-variation tuples, i.e., $\{(\mu_l, \Delta_l) = ([x_q^l]_N, [\delta_q^l]_N) \mid \forall l = 1, \cdots, L\}$. As a result, the general form of an association rule in the ARMS is given as the following,

$$\alpha_1 \wedge \alpha_2 \cdots \wedge \alpha_e \cdots \wedge \alpha_E \Rightarrow \beta_1 \wedge \beta_2 \cdots \wedge \beta_f \cdots \wedge \beta_F \quad [Support = s\%; Confidence = c\%], \tag{21}$$

where $\exists \alpha_e \in \{a_{ij}^*\}_{\sum_{i=1}^{M} n_i} \mid \forall e = 1, \cdots, E \le M$, $\exists \beta_f \in \{(x_q^l, \delta_q^l)\}_{N \times L} \mid \forall f = 1, \cdots, F \le N$, and $s\%$ and $c\%$ refer to the support and confidence levels for this rule, respectively. They are calculated based on the following,

$$s\% = \frac{count(\alpha_1 \wedge \alpha_2 \cdots \wedge \alpha_E \wedge \beta_1 \wedge \beta_2 \cdots \wedge \beta_F)}{count(DB)} \times 100\%, \tag{22}$$

$$c\% = \frac{count(\alpha_1 \wedge \alpha_2 \cdots \wedge \alpha_E \wedge \beta_1 \wedge \beta_2 \cdots \wedge \beta_F)}{count(\alpha_1 \wedge \alpha_2 \cdots \wedge \alpha_E)} \times 100\%, \tag{23}$$

where $count(\alpha_1 \wedge \alpha_2 \cdots \wedge \alpha_E \wedge \beta_1 \wedge \beta_2 \cdots \wedge \beta_F)$ is the number of transaction records in DB containing all items α_1, α_2,…, and α_E as well as β_1, β_2,…, and β_F, $count(DB)$ is the total number of data records contained in DB, and $count(\alpha_1 \wedge \alpha_2 \cdots \wedge \alpha_E)$ is the number of transaction records in DB containing all items α_1, α_2,…, and α_E. In general,

$count(DB) = S$, because each TID corresponds to a $s-t$ pair. In addition, the set $\{\alpha_1, \alpha_2, \cdots, \alpha_e, \cdots, \alpha_E\}$ embodies a non-empty subset of $\{a_{ij}^* \mid \forall i \in [1, M], \exists j \in [1, n_i]\}$, whereas the set $\{\beta_1, \beta_2, \cdots, \beta_f, \cdots, \beta_F\}$ exhibits a non-empty subset of $\{(x_q^l, \delta_q^l) \mid \forall q \in [1, N], \exists l \in [1, L]\}$. The association rule in Eq. (21) means that the data occurrence of α_1, α_2,…, and α_E will most likely (at a $s\%$ -support and with a $c\%$ - confidence) associate with the data occurrence of β_1, β_2,…, and β_F .

A good number of efficient algorithms for mining association rules have been proposed (Chen et al., 1996). The ARMS adopts a well-known algorithm, called Apriori algorithm (Agrawal and Srikant, 1994) to determine frequent itemsets. The idea driving Apriori algorithm is to use an iterative approach known as a level-wise search, where k -itemsets (the itemsets that containing k items) are used to explore $(k+1)$ -itemsets. Apriori property, where all nonempty subsets of a frequent itemset must also be frequent, helps reduce the search space and improve the efficiency of the level-wise generation of frequent itemsets. Once the frequent itemsets are identified from DB, it is straightforward to generate strong association rules from them. For a large volume of source relations, the performance of rule generation may be slow. Rather than updating the association rule base continuously, the ARMS derives association rules incrementally by storing the record counts of previous computing data into the existing rule set and adding the new record counts during the new data computing process. Table 4-1 shows the procedure of such an incremental strategy for rule mining.

Table 4-1 Algorithm of incremental mining of association rules in the ARMS

Begin
Let $N = count(DB)$; /* Total data record count */
Let $S_m = min\_sup$; /* Minimum support threshold specified by the user */
Let $C_m = min\_conf$; /* Minimum confidence threshold specified by the user */
    For $i = 1$ *to* N *do*
        Begin
        Let $S = count(\alpha_1 \wedge \alpha_2 \cdots \wedge \alpha_E \wedge \beta_1 \wedge \beta_2 \cdots \wedge \beta_F)$; /* Call the Apriori algorithm */
        Let $C = count(\alpha_1 \wedge \alpha_2 \cdots \wedge \alpha_E)$; /* Call the Apriori algorithm */
        Let $s = (S/N) \times 100\%$;
        Let $c = (S/C) \times 100\%$;
        If $s \geq S_m$ *and* $c \geq C_m$
                Then $Rule_i$ *is derived;*
        End if;
    End;
End;

4.3.4. Rule Evaluation and Presentation

Based on all the association rules created, the evaluation and presentation module comes into play to refine these rules in order to keep the most relevant and valuable rules in the knowledge base in the form of either case bases or rule bases. The characteristics of each FR cluster should also be explored based on the rules and the related support and confidence levels. Moreover, the causality of original association rules are defined for single feature options, as the precedent of each rule is a subset of $\{a_{ij}^*\}$ and the consequence of each rule is a subset of $\{(x_q^l, \delta_q^l)\}$ per se. Nevertheless, inference relationships do exist in various combinations of more feature options. This means a need for generating combinatorial rules. Finally, users can retrieve all the rules stored in the knowledge base to understand the mappings of CNs to FRs clearly, to gain insights into the consequences of diverse customer preferences on the product fulfillment, and thus to justify the proper specification of product offerings in a portfolio.

4.4. Case Study

The potential of ARMS has been tested in an electronics company that produces a large variety of vibration motors for major world-leading mobile phone manufacturers. The company had conducted extensive market studies and derived data of customer expressions of various functionality related to mobile phones. These data have been collected from market surveys and analyzed based on natural language processing. As far as the "Alarm" function is concerned, the related features and their options are summarized in Table 4-2. Those CNs listed in Table 4-2 provide the ground for diverse specifications of the "Alarm" function as perceived by different mobile phone users. A variety of the "Alarm" functions correspond to different vibration motor designs. In other words, the "Alarm"- related CNs of mobile phones are fulfilled by the FRs of vibration motors. Based on existing product documentation and consultation with design engineers, we know that the functional specification of vibration motors is described by a set of FRs and their values, as shown in Table 4-3. Among these 9 FRs, the "Pbfree" is of binary type and the "Coating" is of nominal type, while all the rest are numerical variables.

It is interesting to observe the difference between CNs and FRs in this case. What customers really perceive is how they feel about the "Alarm" function of mobile phones. Customers have no idea of the implications of this functionality in engineering – vibration motors. From the company's viewpoint, CNs refer to mobile phones, whereas FRs are related to vibration motors. When the company makes decisions about its vibration motor portfolio, it has to understand the mapping mechanisms between the customer and functional domains, as well as the tradeoffs of requirement specification between mobile phones and vibration motors.

Table 4-2 List of CNs

Feature		Option		
$a_i \mid \forall i = 1,\cdots,M$	**Description**	$a_{ij}^* \mid \forall j = 1,\cdots,n_i$	**Code**	**Description**
a_1	Feel of vibration	a_{11}^*	A11	Feel the vibration very strongly
		a_{12}^*	A12	Alarmed by vibration without vibrating suddenly
		a_{13}^*	A13	Sensitive to the vibration
a_2	Price	a_{21}^*	A21	Buy an expensive mobile phone with desire for a long time use
		a_{22}^*	A22	Catch up the mobile phone style occasionally at a low price
		a_{23}^*	A23	Try latest fashion of mobile phones at a moderate price
a_3	Size	a_{31}^*	A31	Portable
		a_{32}^*	A32	Comfortable to hold
		a_{33}^*	A33	Not easy to lose
a_4	Volume of sound	a_{41}^*	A41	Little noise
		a_{42}^*	A42	Alarmed independent of vibration
		a_{43}^*	A43	Alarmed by both vibration and sound
a_5	Material	a_{51}^*	A51	Green material for environment friendliness
a_6	Weight	a_{61}^*	A61	As light as possible

Table 4-3 List of FRs

FR			FR Value		
$v_q \mid \forall q = 1,\cdots,N$	**Description**	**Type**	$v_{qr}^* \mid \forall r = 1,\cdots,n_q$	**Code**	**Description**
v_1	Current	Numerical	v_{11}^*	V11	100 mA
			v_{12}^*	V12	80 mA
			v_{13}^*	V13	60 mA
v_2	Pbfree	Binary	v_{21}^*	V21	1 (Yes)
			v_{22}^*	V22	0 (No)
v_3	Length	Numerical	v_{31}^*	V31	8 mm
			v_{32}^*	V32	12 mm
			v_{33}^*	V33	10 mm
v_4	Diameter	Numerical	v_{41}^*	V41	5 mm
			v_{42}^*	V42	4 mm
			v_{43}^*	V43	6 mm
v_5	Coating	Nominal	v_{51}^*	V51	Au
			v_{52}^*	V52	Alloy
			v_{53}^*	V53	None
v_6	Angle	Numerical	v_{61}^*	V61	40°
			v_{62}^*	V62	55°
v_7	Strength	Numerical	v_{71}^*	V71	7 Kg
			v_{72}^*	V72	4 Kg
v_8	Weight	Numerical	v_{81}^*	V81	2 g
			v_{82}^*	V82	3 g
v_9	Hardness	Numerical	v_{91}^*	V91	40 HB
			v_{92}^*	V92	70 HB

Based on sales records, target data is identified and organized into a transaction database, as shown in Table 4-4. For illustrative simplicity, only 30 out of hundreds of transaction records are used in the case study here. As shown in Table 4-4, each customer order indicates the customer's choice of certain feature options related to the "Alarm" function of mobile phones, which is presented as a specific instance of a subset of $A = \{a_i\}_M$. Corresponding to the 30 customers (end-users of mobile phones), there are 30 vibration motors provided, whose requirement information are described as particular instances of FR vector, $[v_{qr}^*]_N$.

Table 4-4 Transaction database

Record (TID)	**CNs** ($\overline{a_s^*} \mid \forall s = 1,\cdots,S$)	**FRs** ($\overline{v_t^*} \mid \forall t = 1,\cdots,T$)
T001	A11, A21, A31, A43, A51, A61	V11, V21, V31, V42, V53, V62, V71, V82, V92
T002	A11, A21, A43, A51	V11, V21, V31, V41, V51, V61, V71, V81, V92
T003	A12, A22, A33, A61	V12, V21, V33, V43, V51, V61, V72, V82, V91
…	…	…
T028	A13, A22, A33, A41, A61	V13, V22, V31, V42, V52, V61, V72, V81, V91
T029	A11, A21, A31, A43, A51, A61	V12, V22, V33, V43, V52, V62, V72, V81, V92
T030	A12, A22, A33, A42, A61	V11, V22, V33, V42, V53, V61, V72, V82, V91

To prioritize 9 FR variables, the AHP is applied. A 9-scale rating system is used to provide subjective judgments of preference, as shown in Table 4-5. The result of each weight associated with each FR variable is given in Table 4-6.

Table 4-5 Scale for subjective judgment

Verbal judgment of preference	**Numerical rating**
Extremely preferred	1
Very strong to extremely	2
Very strongly preferred	3
Strongly to very strongly	4
Strongly preferred	5
Moderately to strongly	6
Moderately preferred	7
Equally to moderately	8
Equally preferred	9

Due to different metrics used for FR variables, all FR instances in Table 4-4 need to be standardized based on the max-min normalization method. After that, the distances between every two FR instances are calculated to quantify the dissimilarity among them. The SPSS software package (SPSS 12.0 for Windows, http://www.spss.com/) is used to obtain the weighted Euclidean distance measures. The 30 records of product specifications are input into the SPSS software for processing, in which the original data are normalized automatically and then the distances are calculated. The pair-wise measures of distances are presented as a 30×30 matrix. Figure 4-6 shows the raw data for distance measures of numerical FR instances before the normalization. The normalized distance measures of numerical FR instances are presented in a matrix form, $\left[N\_d_{numerical}\left(\overline{v_i^*}, \overline{v_j^*}\right)\right]_{30\times 30}$, as shown in Figure 4-7. The results of distance measures for binary and nominal FR instances, $\left[N\_d_{binary}\left(\overline{v_i^*}, \overline{v_j^*}\right)\right]_{30\times 30}$ and $\left[N\_d_{nominal}\left(\overline{v_i^*}, \overline{v_j^*}\right)\right]_{30\times 30}$, are shown in Figures 4-8 and 4-9, respectively. Based on these three distance components, the composite distances are calculated and presented as a dissimilarity matrix, $\left[d\left(\overline{v_i^*}, \overline{v_j^*}\right)\right]_{30\times 30}$, for all FR instances, as shown in Figure 4-10. Based on the relative importance of FR variables, the weights associated with numerical, binary and nominal distance components are determined as $W_{numerical} = w_1 + w_3 + w_4 + w_6 + w_7 + w_8 + w_9 = 0.677$, $W_{binary} = w_2 = 0.304$ and $W_{nominal} = w_5 = 0.019$, respectively.

Table 4-6 Relative importance among FR variables

FR (v_q)	**Weight** (w_q)
v_1	0.219
v_2	0.304
v_3	0.046
v_4	0.031
v_5	0.019
v_6	0.066
v_7	0.157
v_8	0.095
v_9	0.083
	$\sum w_q = 1$

Output4 for distance (numeric variables) - SPSS Viewer

File Edit View Insert Format Analyze Graphs Utilities Window Help

	1	2	3	4	5	6	7	8	9	10	11
1	.000	3.801	26.679	16.624	9.703	24.249	.000	6.028	22.652	9.703	7.8
2	3.801	.000	27.826	13.587	14.268	25.395	3.801	5.646	23.798	14.268	4.0
3	26.679	27.826	.000	10.056	13.558	9.931	26.679	29.017	4.028	13.558	23.7
4	16.624	13.587	10.056	.000	15.740	11.931	16.624	22.652	6.028	7.685	9.5
5	9.703	14.268	13.558	15.740	.000	15.433	9.703	19.914	17.585	8.056	18.2
6	24.249	25.395	9.931	11.931	15.433	.000	24.249	26.586	5.903	15.433	29.4
7	.000	3.801	26.679	16.624	9.703	24.249	.000	6.028	22.652	9.703	7.8
8	6.028	5.646	29.017	22.652	19.914	26.586	6.028	.000	24.989	19.914	9.6
9	22.652	23.798	4.028	6.028	17.585	5.903	22.652	24.989	.000	9.530	19.7
10	9.703	14.268	13.558	7.685	8.056	15.433	9.703	19.914	9.530	.000	10.2
11	7.828	4.028	23.798	9.560	18.296	29.423	7.828	9.674	19.770	10.240	.0
12	7.964	11.765	18.715	24.588	9.612	24.340	7.964	6.119	22.743	17.668	15.7
13	19.460	19.842	3.801	9.674	13.940	13.731	19.460	17.615	7.828	13.940	15.8
14	7.912	11.712	18.768	16.767	9.560	24.393	7.912	13.940	22.795	17.615	15.7
15	7.828	4.028	23.798	9.560	18.296	29.423	7.828	9.674	19.770	10.240	.0
16	7.468	7.850	15.793	5.737	10.003	9.612	7.468	13.496	11.765	10.003	11.8
17	17.030	17.412	13.731	11.549	15.815	3.801	17.030	15.185	9.703	15.815	21.4
18	13.237	13.619	17.524	15.524	3.966	15.649	13.237	19.265	21.552	12.022	17.6
19	13.814	17.615	12.865	10.865	11.712	22.795	13.814	19.842	16.893	11.712	13.5
20	13.496	18.061	9.765	11.765	11.848	3.584	13.496	15.834	5.737	11.848	22.0
21	17.412	25.395	16.767	18.768	15.433	6.837	17.412	19.749	12.740	15.433	29.4
22	20.560	17.524	6.119	3.937	19.677	7.994	20.560	18.715	2.091	11.621	13.4
23	11.496	11.878	11.765	9.765	5.975	13.640	11.496	17.524	15.793	14.031	15.9
24	11.856	8.056	19.770	13.587	14.268	33.451	11.856	13.701	23.798	14.268	4.0
25	22.301	19.265	11.878	9.696	17.668	10.003	22.301	20.456	7.850	9.612	15.2
26	11.878	19.861	14.801	8.928	13.649	16.676	11.878	22.089	10.774	5.593	15.8
27	22.249	19.212	11.931	9.931	9.560	10.056	22.249	28.277	15.959	17.615	23.2
28	13.640	9.839	17.986	19.677	12.302	19.861	13.640	7.612	22.014	20.358	13.8
29	14.801	11.765	11.878	17.751	9.612	17.503	14.801	12.956	15.906	17.668	15.7
30	12.596	9.560	14.084	4.028	11.712	7.903	12.596	18.624	10.056	11.712	13.5

This is a dissimilarity matrix

Figure 4-6 Raw data for distance measures of numerical FR instances

$$
\left[\begin{array}{*{30}{c}}
0 \\
.076 & 0 \\
.534 & .557 & 0 \\
.332 & .272 & .2 & 0 \\
.194 & .285 & .271 & .315 & 0 \\
.485 & .508 & .2 & .24 & .309 & 0 \\
0 & .076 & .534 & .332 & .194 & .485 & 0 \\
.121 & .113 & .58 & .453 & .398 & .532 & .12 & 0 \\
.453 & .476 & .08 & .121 & .352 & .118 & .453 & .5 & 0 \\
.194 & .285 & .271 & .154 & .161 & .309 & .194 & .398 & .19 & 0 \\
.157 & .08 & .476 & .191 & .366 & .588 & .157 & .193 & .4 & .2 & 0 \\
.16 & .235 & .374 & .492 & .192 & .487 & .159 & .122 & .455 & .353 & .316 & 0 \\
.39 & .397 & .076 & .193 & .279 & .275 & .39 & .352 & .157 & .279 & .316 & .23 & 0 \\
.158 & .234 & .375 & .335 & .191 & .488 & .158 & .279 & .456 & .352 & .315 & .156 & .23 & 0 \\
.157 & .081 & .476 & .191 & .366 & .588 & .157 & .193 & .396 & .2 & 0 & .316 & .316 & .315 & 0 \\
.149 & .157 & .316 & .115 & .2 & .192 & .149 & .27 & .235 & .2 & .238 & .309 & .24 & .152 & .238 & 0 \\
.341 & .348 & .275 & .231 & .316 & .076 & .341 & .304 & .194 & .316 & .429 & .342 & .2 & .343 & .43 & .116 & 0 \\
.265 & .272 & .35 & .31 & .079 & .313 & .265 & .385 & .431 & .24 & .353 & .263 & .274 & .262 & .353 & .196 & .237 & 0 \\
.277 & .352 & .257 & .217 & .234 & .456 & .276 & .397 & .338 & .234 & .272 & .274 & .113 & .118 & .272 & .195 & .312 & .23 & 0 \\
.27 & .361 & .195 & .235 & .237 & .072 & .27 & .317 & .115 & .237 & .442 & .272 & .2 & .273 & .442 & .12 & .08 & .316 & .316 & 0 \\
.348 & .508 & .335 & .375 & .309 & .137 & .348 & .395 & .255 & .309 & .588 & .27 & .275 & .35 & .588 & .192 & .076 & .313 & .319 & .07 & 0 \\
.411 & .35 & .122 & .079 & .394 & .16 & .411 & .374 & .042 & .232 & .27 & .413 & .115 & .414 & .27 & .913 & .152 & .389 & .296 & .157 & .297 & 0 \\
.23 & .238 & .235 & .195 & .12 & .273 & .23 & .35 & .316 & .28 & .318 & .228 & .16 & .072 & .318 & .08 & .197 & .115 & .115 & .2 & .273 & .274 & 0 \\
.237 & .161 & .395 & .272 & .285 & .669 & .237 & .274 & .476 & .285 & .08 & .235 & .236 & .234 & .08 & .318 & .509 & .272 & .191 & .522 & .669 & .35 & ..238 & 0 \\
.446 & .385 & .238 & .194 & .353 & .2 & .446 & .409 & .157 & .192 & .305 & .448 & .23 & .604 & .305 & .309 & .192 & .274 & .411 & .272 & .337 & .115 & .389 & .385 & 0 \\
.238 & .397 & .296 & .179 & .273 & .334 & .238 & .442 & .215 & .112 & .317 & .4 & .235 & .24 & .371 & .157 & .273 & .352 & .122 & .193 & .197 & .257 & .237 & .397 & .372 & 0 \\
.445 & .384 & .239 & .2 & .191 & .2 & .445 & .566 & .319 & .352 & .465 & .443 & .23 & .287 & .465 & .152 & .193 & .112 & .255 & .273 & .338 & .277 & .07 & .384 & .318 & .377 & 0 \\
.273 & .197 & .36 & .394 & .246 & .397 & .273 & .152 & .44 & .407 & .277 & .114 & .2 & .27 & .277 & .279 & .238 & .158 & .313 & .326 & .397 & .315 & .198 & .197 & .275 & .519 & .27 & 0 \\
.296 & .235 & .238 & .355 & .192 & .35 & .296 & .259 & .318 & .353 & .316 & .137 & .23 & .293 & .316 & .309 & .342 & .263 & .411 & .272 & .487 & .276 & .228 & .235 & .311 & .534 & .306 & .114 & 0 \\
.252 & .191 & .282 & .081 & .234 & .158 & .252 & .372 & .2 & .234 & .272 & .411 & .274 & .255 & .272 & .03 & .15 & .23 & .298 & .155 & .295 & .159 & .115 & .352 & .274 & .259 & .118 & .313 & .274 & 0
\end{array}\right]_{30\times30}
$$

Figure 4-7 Result of distance measures for numerical FR instances

$$
\begin{bmatrix}
0 \\
0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}_{30\times 30}
$$

Figure 4-8 Result of distance measures for binary FR instances

$$
\begin{bmatrix}
0 \\
0 & 0 \\
1 & 1 & 0 \\
1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0
\end{bmatrix}_{30\times 30}
$$

Figure 4-9 Result of distance measures for nominal FR instances

$$
\begin{bmatrix}
0 \\
.076 & 0 \\
.554 & .58 & 0 \\
.352 & .29 & .2 & 0 \\
.52 & .6 & .57 & .62 & 0 \\
.51 & .53 & .22 & .26 & .53 & 0 \\
.3 & .38 & .85 & .65 & .22 & .81 & 0 \\
.12 & .11 & .6 & .47 & .82 & .55 & .42 & 0 \\
.47 & .5 & .08 & .12 & .65 & .14 & .77 & .52 & 0 \\
.22 & .31 & .29 & .17 & .48 & .31 & .52 & .42 & .21 & 0 \\
.48 & .4 & .78 & .49 & .37 & .91 & .18 & .51 & .7 & .52 & 0 \\
.16 & .24 & .39 & .51 & .51 & .51 & .46 & .12 & .48 & .37 & .64 & 0 \\
.41 & .42 & .08 & .19 & .58 & .3 & .71 & .37 & .16 & .3 & .62 & .25 & 0 \\
.16 & .24 & .4 & .34 & .51 & .51 & .46 & .28 & .48 & .37 & .64 & .26 & .25 & 0 \\
.46 & .38 & .74 & .51 & .39 & .91 & .16 & .49 & .72 & .52 & .02 & .62 & .64 & .62 & 0 \\
.17 & .18 & .32 & .12 & .5 & .21 & .47 & .29 & .24 & .22 & .54 & .33 & .24 & .17 & .44 & 0 \\
.36 & .37 & .3 & .25 & .64 & .08 & .66 & .32 & .21 & .32 & .75 & .36 & .22 & .36 & .75 & .14 & 0 \\
.59 & .6 & .67 & .63 & .01 & .62 & .29 & .71 & .75 & .54 & .37 & .58 & .6 & .58 & .37 & .52 & .54 & 0 \\
.3 & .32 & .26 & .22 & .53 & .48 & .6 & .42 & .34 & .25 & .57 & .29 & .11 & .14 & .6 & .2 & .33 & .55 & 0 \\
.29 & .38 & .2 & .24 & .54 & .09 & .59 & .34 & .12 & .26 & .74 & .29 & .2 & .29 & .76 & .12 & .1 & .64 & .32 & 0 \\
.27 & .53 & .36 & .4 & .53 & .14 & .67 & .42 & .28 & .31 & .91 & .29 & .3 & .37 & .91 & .21 & .01 & .63 & .34 & .09 & 0 \\
.43 & .37 & .12 & .08 & .69 & .18 & .73 & .39 & .04 & .25 & .57 & .43 & .12 & .43 & .59 & .93 & .15 & .69 & .3 & .16 & .32 & 0 \\
.25 & .26 & .24 & .2 & .42 & .25 & .55 & .37 & .32 & .3 & .62 & .24 & .16 & .09 & .64 & .08 & .22 & .44 & .12 & .2 & .29 & .27 & 0 \\
.24 & .16 & .42 & .29 & .61 & .69 & .44 & .27 & .5 & .31 & .4 & .24 & .26 & .23 & .38 & .34 & .53 & .59 & .21 & .54 & .69 & .37 & .27 & 0 \\
.77 & .7 & .56 & .51 & .37 & .5 & .47 & .73 & .48 & .49 & .23 & .77 & .55 & .62 & .23 & .63 & .49 & .27 & .73 & .6 & .64 & .44 & .71 & .71 & 0 \\
.56 & .72 & .62 & .5 & .29 & .54 & .26 & .76 & .54 & .41 & .34 & .72 & .56 & .56 & .34 & .48 & .57 & .35 & .44 & .51 & .5 & .58 & .56 & .72 & .69 & 0 \\
.47 & .4 & .26 & .22 & .51 & .2 & .77 & .59 & .34 & .35 & .79 & .46 & .25 & .25 & .79 & .17 & .19 & .41 & .28 & .29 & .34 & .3 & .09 & .4 & .34 & .68 & 0 \\
.29 & .22 & .36 & .39 & .55 & .42 & .59 & .17 & .44 & .43 & .58 & .13 & .2 & .2 & .6 & .28 & .26 & .48 & .31 & .33 & .42 & .32 & .2 & .22 & .3 & .84 & .29 & 0 \\
.3 & .24 & .26 & .38 & .51 & .37 & .6 & .26 & .34 & .37 & .54 & .14 & .25 & .25 & .62 & .33 & .36 & .58 & .43 & .29 & .51 & .3 & .25 & .24 & .31 & .85 & .23 & .13 & 1 \\
.27 & .21 & .28 & .08 & .53 & .16 & .57 & .39 & .2 & .25 & .57 & .43 & .27 & .27 & .59 & .03 & .17 & .55 & .3 & .16 & .32 & .16 & .18 & .37 & .29 & .58 & .14 & .31 & .29 & 0
\end{bmatrix}_{30 \times 30}
$$

Figure 4-10 Dissimilarity matrix based on distance measures for all FR instances

Based on the dissimilarity matrix, a fuzzy compatible matrix, R, is determined, as shown in Figure 4-11. Obviously, R meets both the reflexive and symmetric characteristics. To obtain a fuzzy equivalence matrix, the max-min composition is applied. The result of $R^2 = R \circ R$ is shown in Figure 4-12. As $R^2 \neq R$, we know R^2 is not a fuzzy equivalence matrix yet. Continuing to apply the max-min composition, $R^4 = (R \circ R) \circ (R \circ R)$ is obtained, which equals to R^2. The result of R^4 is also shown in Figure 4-12. As a result, R^4 turns out to be a fuzzy equivalence matrix. Based on R^4, the λ-cut is derived with a similarity threshold setting at 0.84. The result of the λ-cut is shown in Figure 4-13.

With the obtained λ-cut, a fuzzy netting graph is constructed, as shown in Figure 4-14. Based on the partitions derived from the fuzzy netting graph, 3-clusters of FR instances are identified. The mean value and variation range for each FR cluster are calculated based on those FR instances that are grouped into this cluster. The result of FR clustering is given in

Table 4-7, in which, for example, FR cluster, χ_1, is associated with its mean, $\mu_1 = [100, Y, 9.2, 4.5, Au, 44.5, 6.7, 2.4, 49]$, and variation range, $\delta_1 = [0, 0, 1.2, 0.5, 0, 10.5, 2.7, 0.6, 21]$, and contains 10 FR instances, including $\overline{v_1^*}$, $\overline{v_2^*}$, $\overline{v_7^*}$, $\overline{v_8^*}$, $\overline{v_{11}^*}$, $\overline{v_{12}^*}$, $\overline{v_{14}^*}$, $\overline{v_{15}^*}$, $\overline{v_{24}^*}$, and $\overline{v_{29}^*}$.

$$\left[\begin{array}{cccccccccccccccccccccccccccccc}
1 \\
.924 & 1 \\
.446 & .42 & 1 \\
.648 & .71 & .8 & 1 \\
.48 & .4 & .43 & .38 & 1 \\
.49 & .47 & .78 & .74 & .47 & 1 \\
.7 & .62 & .15 & .35 & .78 & .19 & 1 \\
.88 & .89 & .4 & .53 & .18 & .45 & .58 & 1 \\
.53 & .5 & .92 & .88 & .35 & .86 & .23 & .48 & 1 \\
.78 & .69 & .71 & .83 & .52 & .69 & .48 & .58 & .79 & 1 \\
.52 & .6 & .22 & .51 & .63 & .09 & .82 & .49 & .3 & .48 & 1 \\
.84 & .76 & .61 & .49 & .49 & .49 & .54 & .88 & .52 & .63 & .36 & 1 \\
.59 & .58 & .92 & .81 & .42 & .7 & .29 & .63 & .84 & .7 & .38 & .75 & 1 \\
.84 & .76 & .6 & .66 & .49 & .49 & .54 & .72 & .52 & .63 & .36 & .74 & .75 & 1 \\
.54 & .62 & .26 & .49 & .61 & .09 & .84 & .51 & .28 & .48 & .98 & .38 & .36 & .38 & 1 \\
.83 & .82 & .68 & .88 & .5 & .79 & .53 & .71 & .76 & .78 & .46 & .67 & .76 & .83 & .56 & 1 \\
.64 & .63 & .7 & .75 & .36 & .92 & .34 & .68 & .79 & .68 & .25 & .64 & .78 & .64 & .25 & .86 & 1 \\
.41 & .4 & .33 & .37 & .99 & .38 & .71 & .29 & .25 & .46 & .63 & .42 & .4 & .42 & .63 & .48 & .46 & 1 \\
.7 & .68 & .74 & .78 & .47 & .52 & .4 & .58 & .66 & .75 & .43 & .71 & .89 & .86 & .4 & .8 & .67 & .45 & 1 \\
.71 & .62 & .8 & .76 & .46 & .91 & .41 & .66 & .88 & .74 & .26 & .71 & .8 & .71 & .24 & .88 & .9 & .36 & .68 & 1 \\
.73 & .47 & .64 & .6 & .47 & .86 & .33 & .58 & .72 & .69 & .09 & .71 & .7 & .63 & .09 & .79 & .99 & .37 & .66 & .91 & 1 \\
.57 & .63 & .88 & .92 & .31 & .82 & .27 & .61 & .96 & .75 & .43 & .57 & .88 & .57 & .41 & .07 & .85 & .31 & .7 & .84 & .68 & 1 \\
.75 & .74 & .76 & .8 & .58 & .75 & .45 & .63 & .68 & .7 & .38 & .76 & .84 & .91 & .36 & .92 & .78 & .56 & .88 & .8 & .71 & .73 & 1 \\
.76 & .84 & .58 & .71 & .39 & .31 & .56 & .73 & .5 & .69 & .6 & .76 & .74 & .77 & .62 & .66 & .47 & .41 & .79 & .46 & .31 & .63 & .73 & 1 \\
.23 & .3 & .44 & .49 & .63 & .5 & .53 & .27 & .52 & .51 & .77 & .23 & .45 & .38 & .77 & .37 & .51 & .73 & .27 & .4 & .36 & .56 & .29 & .29 & 1 \\
.44 & .28 & .38 & .5 & .71 & .46 & .74 & .24 & .46 & .59 & .66 & .28 & .44 & .44 & .66 & .52 & .43 & .65 & .56 & .49 & .5 & .42 & .44 & .28 & .31 & 1 \\
.53 & .6 & .74 & .78 & .49 & .8 & .23 & .41 & .66 & .65 & .21 & .54 & .75 & .75 & .21 & .83 & .81 & .59 & .72 & .71 & .66 & .7 & .91 & .6 & .66 & .32 & 1 \\
.71 & .78 & .64 & .61 & .45 & .58 & .41 & .83 & .56 & .57 & .42 & .87 & .8 & .8 & .4 & .72 & .74 & .52 & .69 & .67 & .58 & .68 & .8 & .78 & .7 & .16 & .71 & 1 \\
.7 & .76 & .74 & .62 & .49 & .63 & .4 & .74 & .66 & .63 & .46 & .86 & .75 & .75 & .38 & .67 & .64 & .42 & .57 & .71 & .49 & .7 & .75 & .76 & .69 & .15 & .77 & .87 & 1 \\
.73 & .79 & .72 & .92 & .47 & .84 & .43 & .61 & .8 & .75 & .43 & .57 & .73 & .73 & .41 & .97 & .83 & .45 & .7 & .84 & .68 & .84 & .82 & .63 & .71 & .42 & .86 & .69 & .71 & 1
\end{array}\right]_{30\times 30}$$

Figure 4-11 Result of R

$$\left[\begin{array}{cccccccccccccccccccccccccccccc}
1 \\
.7 & 1 \\
.64 & .74 & 1 \\
.52 & .82 & .8 & 1 \\
.7 & .62 & .85 & .85 & 1 \\
.75 & .71 & .76 & .76 & .78 & 1 \\
.7 & .7 & .56 & .648 & .78 & .79 & 1 \\
.88 & .88 & .73 & .71 & .58 & .71 & .9 & 1 \\
.76 & .76 & .92 & .88 & .58 & .76 & .53 & .71 & 1 \\
.78 & .78 & .8 & .83 & .58 & .79 & .7 & .78 & .85 & 1 \\
.7 & .62 & .58 & .6 & .78 & .51 & .82 & .6 & .52 & .6 & 1 \\
.84 & .84 & .82 & .75 & .58 & .75 & .7 & .78 & .75 & .78 & .6 & 1 \\
.76 & .75 & .92 & .8 & .52 & .78 & .59 & .75 & .82 & .81 & .8 & .75 & 1 \\
.84 & .74 & .8 & .75 & .58 & .79 & .7 & .84 & .76 & .78 & .6 & .84 & .76 & 1 \\
.7 & .62 & .58 & .62 & .78 & .56 & .84 & .62 & .53 & .78 & .82 & .62 & .62 & .62 & 1 \\
.83 & .83 & .8 & .78 & .58 & .89 & .7 & .83 & .78 & .83 & .83 & .83 & .81 & .83 & .62 & 1 \\
.68 & .82 & .79 & .79 & .58 & .72 & .64 & .71 & .86 & .79 & .81 & .76 & .75 & .83 & .62 & .79 & 1 \\
.7 & .62 & .56 & .49 & .79 & .89 & .78 & .58 & .59 & .56 & .71 & .65 & .59 & .59 & .73 & .59 & .59 & 1 \\
.74 & .8 & .89 & .81 & .58 & .79 & .7 & .73 & .84 & .78 & .72 & .76 & .78 & .75 & .62 & .83 & .8 & .59 & 1 \\
.83 & .82 & .8 & .8 & .58 & .81 & .7 & .71 & .8 & .76 & .6 & .75 & .8 & .75 & .62 & .79 & .91 & .59 & .8 & 1 \\
.79 & .79 & .78 & .79 & .58 & .86 & .7 & .73 & .82 & .78 & .68 & .73 & .78 & .79 & .56 & .86 & .76 & .59 & .71 & .76 & 1 \\
.77 & .71 & .78 & .8 & .63 & .86 & .62 & .73 & .78 & .83 & .7 & .75 & .78 & .75 & .62 & .88 & .82 & .59 & .78 & .78 & .82 & 1 \\
.83 & .82 & .8 & .8 & .58 & .79 & .7 & .76 & .8 & .8 & .71 & .76 & .8 & .81 & .62 & .83 & .78 & .58 & .78 & .88 & .79 & .8 & 1 \\
.84 & .84 & .74 & .71 & .61 & .73 & .7 & .74 & .74 & .76 & .88 & .76 & .75 & .76 & .62 & .82 & .74 & .62 & .79 & .74 & .73 & .74 & .79 & 1 \\
.71 & .71 & .71 & .71 & .63 & .56 & .77 & .7 & .74 & .74 & .76 & .88 & .76 & .75 & .76 & .62 & .82 & .74 & .62 & .79 & .74 & .73 & .74 & .79 & 1 \\
.7 & .62 & .59 & .56 & .71 & .59 & .74 & .58 & .89 & .59 & .74 & .59 & .56 & .59 & .74 & .59 & .59 & .71 & .59 & .59 & .59 & .59 & .59 & .62 & .66 & 1 \\
.83 & .75 & .78 & .78 & .6 & .8 & .6 & .71 & .8 & .78 & .6 & .75 & .78 & .75 & .6 & .79 & .8 & .89 & .78 & .8 & .8 & .8 & .83 & .74 & .59 & .59 & 1 \\
.78 & .83 & .8 & .81 & .58 & .75 & .7 & .78 & .8 & .72 & .7 & .76 & .8 & .76 & .62 & .88 & .78 & .73 & .8 & .8 & .71 & .8 & .8 & .78 & .7 & .57 & .75 & 1 \\
.84 & .76 & .75 & .75 & .58 & .74 & .7 & .76 & .74 & .71 & .6 & .76 & .75 & .76 & .62 & .76 & .75 & .69 & .75 & .75 & .71 & .75 & .76 & .76 & .71 & .59 & .75 & .86 & 1 \\
.83 & .82 & .85 & .92 & .58 & .8 & .7 & .79 & .88 & .83 & .71 & .76 & .81 & .76 & .62 & .78 & .84 & .71 & .78 & .84 & .82 & .82 & .8 & .79 & .71 & .59 & .8 & .78 & .76 & 1
\end{array}\right]_{30\times 30}$$

Figure 4-12 Result of R^2 and R^4

$$
\begin{bmatrix}
1 \\
0 & 1 \\
0 & 0 & 1 \\
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}_{30\times 30}
$$

Figure 4-13 Result of a λ-cut with $\lambda = 0.84$

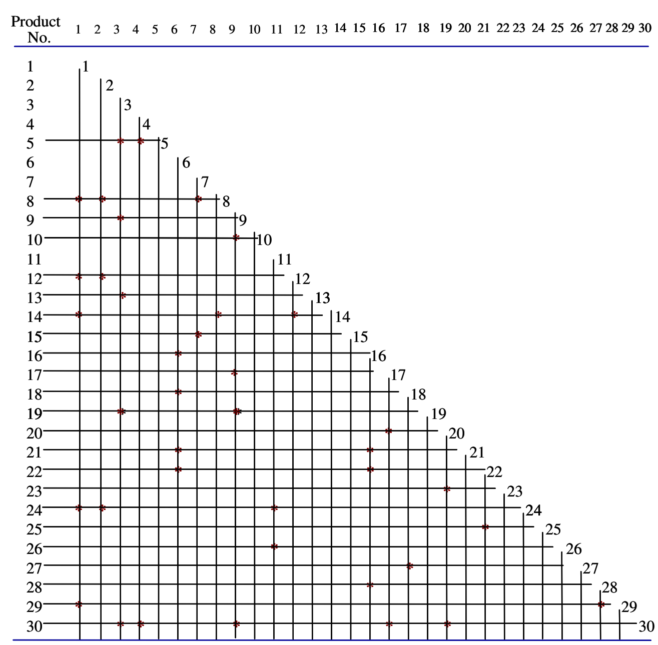


Figure 4-14 Fuzzy netting graph

Table 4-7 Result of FR clustering

FR Cluster			Clustered FR Instances ($\{\overline{v_t^*} \sim \chi_l \mid \forall t = 1, \cdots, n_l \leq T\}$)
χ_l	Mean Value (μ_l)	Variation Range (Δ_l)	
χ_1	[100,Y,9.2,4.5,Au, 44.5,6.7,2.4,49]	[0, 0, 1.2, 0.5, 0, 10.5, 2.7, 0.6, 21]	$\{\overline{v_1^*}, \overline{v_2^*}, \overline{v_7^*}, \overline{v_8^*}, \overline{v_{11}^*}, \overline{v_{12}^*}, \overline{v_{14}^*}, \overline{v_{15}^*}, \overline{v_{24}^*}, \overline{v_{29}^*}\}$
χ_2	[78.3,Y,11.17,5,Alloy, 47,4.5,2.42,57.5]	[21.7, 0, 1.17, 0.5, 0, 8, 2.5, 0.58, 17.5]	$\{\overline{v_3^*}, \overline{v_4^*}, \overline{v_5^*}, \overline{v_9^*}, \overline{v_{10}^*}, \overline{v_{13}^*}, \overline{v_{17}^*}, \overline{v_{19}^*}, \overline{v_{20}^*}, \overline{v_{23}^*}, \overline{v_{26}^*}, \overline{v_{30}^*}\}$
χ_3	[67.5,Y,10.75,5.13,None, 42.5,5.13,2.38,47.5]	[12.5, 0, 1.25, 0.87, 0, 12.5, 1.87, 0.62, 22.5]	$\{\overline{v_6^*}, \overline{v_{16}^*}, \overline{v_{18}^*}, \overline{v_{21}^*}, \overline{v_{22}^*}, \overline{v_{25}^*}, \overline{v_{27}^*}, \overline{v_{28}^*}\}$

The resulted FR clusters comprise an itemset, $X = \{(x_q^l, \delta_q^l) \mid \forall q \in [1,9]; \exists l \in [1,3]\}$, as shown in Table 4-8. The characteristics of each FR cluster entail the specification of a product platform – a set of base values together with the related variation ranges, and therefore they can be used to suggest standard settings for a vibration motor portfolio. These items are added to the transaction database. The link of each customer order to a FR instance is then replaced with the link to the items of the FR cluster to which this FR instance belongs. For mining rules between itemsets A^* and X, a data mining tool, called Magnum Opus (Version 2.0, http://www.rulequest.com/), is employed. All data are extracted from the transaction database and input as a text file to Magnum Opus. The system allows data to be input as identifier-item files that list customers to be analyzed in the identifier-item format. Each customer has a unique identifier consisting of two columns: one for the identifier and one for the item. Under either search mode, Magnum Opus finds a number of association rules specified by the user. The search guarantees that only those rules with the highest values on the specified metric are found according to user-specified search settings. Magnum Opus will find fewer than the specified number of association rules if the search is terminated by the user or there are fewer than the specified number of associations that satisfy user specified search settings. In this case, the maximum number of associations is set to 10000 to make sure that the association rules can be derived completely, shown in Figure 4-15.

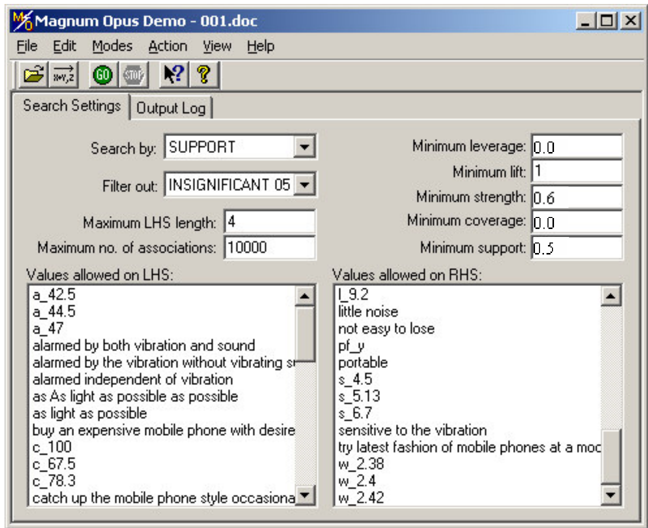


Figure 4-15 Association rule induction in the Magnum Opus

Table 4-8 Specification of vibration motor portfolio based on FR clusters

FR Variable	FR Value	
	Base Value	Variation Range
Current (mA)	100	±0
	78.3	±21.7
	67.5	±12.5
Pbfree	1 (Yes)	±0
Length (mm)	9.2	±1.2
	11.17	±1.17
	10.75	±1.25
Diameter (mm)	4.5	±0.5
	5.5	±0.5
	5.13	±0.87
Coating	Au	±0
	Alloy	±0
	None	±0
Angle (°)	44.5	±10.5
	47	±8
	42.5	±12.5
Strength (Kg)	6.7	±2.7
	4.5	±2.5
	5.13	±1.87
Weight (g)	2.4	±0.6
	2.42	±0.58
	2.38	±0.62
Hardness (HB)	49	±21
	57.5	±17.5
	47.5	±22.5

At the end of rule mining, the system generates 37 association rules, as shown in Table 4-9. These rules serve as the basis of knowledge discovery. Some rules, for example, Rules 31, 32 and 33, are coupled and should be aggregated into one. The possibility of some rule combinations is also considered to discover more implicit rules. For example, Rules 15, 16 and 17 together with Rules 23, 24 and 25 can give more insights to optimize the size of motors. In addition to such rule refinement, the characteristics of each FR cluster and implicit relationships among them are explored to gain more understanding of vibration motor design specifications, so as to identify prominent settings of particular FR variables, to analyze the tradeoffs between different customer perceptions on mobile phones and the relevant FR values of vibration motors, and so on. All the identified patterns of CNs, FRs and the mapping are built into the knowledge base and are utilized to assist users in portfolio decision making based on the generated portfolio (see Table 4-8).

4.5. Sensitivity Analysis

To evaluate the performance of ARMS, the sensitivity of the identified product portfolio is studied with respect to varying values of data mining parameters, including the similarity threshold, and the minimum support and confidence levels. These parameters involve two modules of the ARMS: FR clustering and association rule mining, respectively.

The FR clustering module entails the specification of an optimal value of similarity threshold for the λ-cut. Essentially, it gives rise to a tradeoff issue of FR granularity inherent in mass customization (Tseng and Jiao, 1996). With a large (small) value of the λ-cut, more (fewer) FR clusters will be identified. These FR clusters affect the downstream planning of the product and process platforms. At the economic latitude, the cost of introducing more FRs (i.e., finer FR clustering) and its contribution to customer-perceived values should reach a balance at the right level of aggregation of the product and process platforms. If the differentiation of FRs is too spread or too low a level of aggregation, such as at the nuts and bolts level, then the number of DPs and PVs may be too many, and product fulfillment becomes difficult to leverage investments. To the contrary, if the FR aggregation is at a very high level, such as complete subassemblies, then the repetition may not be sufficient to take advantage of mass production efficiency.

Table 4-9 Result of association rule mining

Rule 1: Green material for environment friendliness\=>pf_y\[Support=0.882; Strength=1.000];
Rule 2: Alarmed independent of vibration\&Not easy to lose\&Catch up the mobile phone style occasionally at a low price\=>h_57.5[±17.5]\[Support=0.265; Strength=0.900];
Rule 3: Alarmed independent of vibration\&Try latest fashion of mobile phones at a moderate price\&Not easy to lose\=> c_78.3[±21.7]\[Support=0.265; Strength=0.900];
Rule 4: Alarmed independent of vibration\&Buy an expensive mobile phone with desire for a long time use \=> l_11.17[±1.17]\[Support=0.265; Strength=0.900];
Rule 5: Alarmed independent of vibration\=>h_57.5[±17.5]\[Support=0.294; Strength=0.833];
Rule 6: Not easy to lose\&Alarmed independent of vibration\=> a_47[±8]\[Support=0.265; Strength=0.900];
Rule 7: Not easy to lose\&Comfortable to hold\&Catch up the mobile phone style occasionally at a low price\=>w_2.42[±0.58]\[Support=0.265; Strength=0.750];
Rule 8: Not easy to lose\=> w_2.42[±0.58]\&h_57.5[±17.5]\[Support=0.265; Strength=0.900];
Rule 9: Catch up the mobile phone style occasionally at a low price\=>co_None\[Support=0.324; Strength=0.688];
Rule 10: Buy an expensive mobile phone with desire for a long time use\&Feel the vibration very strongly\=>h_49[±21]\[Support=0.206; Strength=1.000];
Rule 11: Buy an expensive mobile phone with desire for a long time use\=>s_6.7[±2.7]\[Support=0.206; Strength=1.000];
Rule 12: Buy an expensive mobile phone with desire for a long time use\&Alarmed by both vibration and sound\=>a_44.5[±10.5]\[Support=0.206; Strength=1.000];
Rule 13: Buy an expensive mobile phone with desire for a long time use\&Portable\=>a_44.5[±10.5\[Support=0.265; Strength=0.818];
Rule 14: Buy an expensive mobile phone with desire for a long time use\=>co_Au\[Support=0.206; Strength=1.000];
Rule 15: Feel the vibration very strongly\&Portable\=> l_9.2[±1.2]\[Support=0.206; Strength=0.875];
Rule 16: Feel the vibration very strongly\=>c_100[±0]\[Support=0.206; Strength=0.875];
Rule 17: Feel the vibration very strongly\&As light as possible\=>d_4.5[±0.5]\[Support=0.265; Strength=0.750];
Rule 18: As light as possible\=>a_42.5[±12.5]\[Support=0.206; Strength=0.875];
Rule 19: As light as possible\&Little noise=>w_2.38[±0.62]\[Support=0.206; Strength=0.875];
Rule 20: As light as possible\=>co_None\[Support=0.206 ; Strength=0.875]
Rule 21: Alarmed by the vibration without vibrating suddenly\=>s_4.5[±2.5]\[Support=0.294; Strength=0.833];
Rule 22: Alarmed by the vibration without vibrating suddenly\=>l_11.17[±1.17]\[Support=0.294; Strength=0.833];
Rule 23: Portable\&As light as possible\=>d_4.5[±0.5]\[Support=0.265; Strength=0.818];
Rule 24: Portable\&Feel the vibration very strongly\=>l_9.2[±1.2]\[Support=0.265; Strength=0.818];
Rule 25: Portable\=>a_44.5[±10.5]\[Support=0.294; Strength=0.833];
Rule 26: Sensitive to the vibration\=>d_5.13[±0.87]\[Support=0.235; Strength=0.800];
Rule 27: Sensitive to the vibration\&Little noise\=>c_67.5[±12.5]\[Support=0.235; Strength=0.800];
Rule 28: Sensitive to the vibration\&Little noise\&As light as possible\=>h_47.5[±22.5]\[Support=0.235; Strength=0.727];
Rule 29: Little noise\&As light as possible\=>s_5.13[±1.87]\[Support=0.206; Strength=0.700];
Rule 30: Little noise\=>c_67.5[±12.5]\[Support=0.206; Strength=0.700];
Rule 31: Alarmed by both vibration and sound\=>a_44.5[±10.5]\[Support=0.206; Strength=0.700];
Rule 32: Alarmed by both vibration and sound\=>d_4.5[±0.5]\[Support=0.206; Strength=0.700];
Rule 33: Alarmed by both vibration and sound\=>l_10.75[±1.25]\[Support=0.206; Strength=0.700];
Rule 34: Comfortable to hold\=>w_2.40[±0.6]\[Support=0.206; Strength=0.700];
Rule 35: Try latest fashion of mobile phones at a moderate price\&Alarmed by both vibration and sound\=>c_78.3[±21.7]\[Support=0.206; Strength=0.700];
Rule 36: Try latest fashion of mobile phones at a moderate price\=>d_5.5[±0.5]\[Support=0.206; Strength=0.700];
Rule 37: Try latest fashion of mobile phones at a moderate price\=>co_Alloy\[Support=0.294; Strength=0.833];

An optimal granularity can normally be determined by assessing the performance of the product and process platforms in accordance with the resulting FR clusters. Jiao et al. (2004a) apply real options theory to the valuation of flexibility enabled by the product and process platforms. On the other hand, the construction of the product and process platforms embodies a type of fixed costs (Meyer and Lehnerd, 1997; Du et al., 2001). Therefore, we introduce a performance measure for the λ-cut, Ψ^{λ}, as the following,

$$\Psi^{\lambda} = \frac{E[V]}{C^F}, \tag{24}$$

where $E[V]$ denotes the expected value of the product and process platforms, which is determined based on a real options framework (Jiao et al., 2004b; Gonzalez-Zugasti et al., 2001), and C^F stands for the fixed cost of the product and process platforms. Furthermore, Jiao and Tseng (2004) posit the rationale of justifying cost implications of the product and process platforms based on process variations. Following Jiao and Tseng (2004) and Jiao et al. (2004b), we employ a process capability index to measure the above fixed cost, as the following,

$$C^F = \beta^F e^{\frac{1}{PCI}} = \beta^F e^{\frac{6\sigma}{USL-LSL}}, \tag{25}$$

where β^F is a constant indicating the average dollar cost per variation of process capabilities, USL, LSL and σ are the upper specification limit, lower specification limit and standard deviation of part-worth cost estimates corresponding to individual FR clusters, respectively. The part-worth cost estimates are determined using a pragmatic approach based on standard time estimation (Jiao et al., 2004b).

To analyze the sensitivity of product portfolio identification, a total number of 17 runs of FR clustering are generated by changing λ value from 0.1 to 0.95 with an increment of 0.05. Using process data of vibration motors in Jiao et al. (2003) and flexibility valuation data of vibration motors in Jiao et al. (2004a), the result of sensitivity analysis is obtained. As shown in Figure 4-16, the performance measure in Eq. (24) is presented as a normalized comparison. The result clearly shows that a λ value of 0.84 yields the best performance of FR clustering for product portfolio identification.

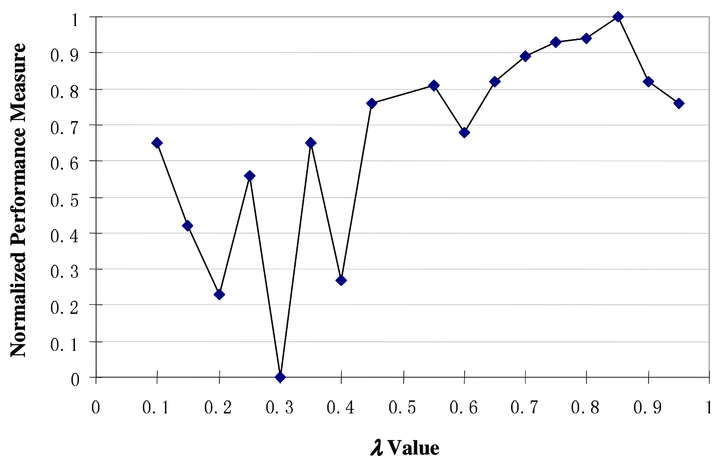


Figure 4-16 Sensitivity analysis of product portfolio identification with respect to similarity threshold

The difficulty in association rule mining originates from the need for determining appropriate thresholds for the support and confidence levels. If the support and confidence thresholds are planned with low values, useful information may be overwhelmed by excessive rules. To the contrary, certain relationship patterns that are of interest may be ignored if the support and confidence criteria are specified too strictly.

Association rules basically suggest the mapping relationships between CNs and FRs. To meet the required CNs, the associated FRs must be fulfilled through configuration of DPs and PVs within the existing product and process platforms – a process of product variant derivation (Du et al., 2001). Such a variant derivation exhibits the accounting of a type of variable costs (Meyer and Lehnerd, 1997). Jiao et al. (2004b) review the implications of customer-perceived value per unit cost in regard to the measure of profitability. Therefore, we introduce a performance measure of association rule mining, ψ^{AR}, based on the ratio of utility to variable cost, as the following,

$$\Psi^{AR} = \sum_{i=1}^{I}\sum_{j=1}^{J}\frac{U_{ij}}{C_j^V}, \tag{26}$$

where the resulted product portfolio comprises $j = 1,\cdots,J$ products that are offered to meet a target market segment with $i = 1,\cdots,I$ customers, U_{ij} denotes the utility of the *i*-th customer with respect to the *j*-th product, and C_j^V is the related variable cost of producing this product variant. As suggested in Jiao et al. (2004b), product level utilities, $\{U_{ij}\}_{I \cdot J}$, are derived from part-worth utilities of individual CNs based on conjoint analysis (Green and Krieger, 1978). Likewise, product costs, $\{C_j^V\}_J$, are determined by the regression of part-worth cost estimates of individual FRs. The association rules indicate what FRs are to be used to satisfy what CNs. Such customer choice and product instantiation can be implemented by introducing binary variables to the part-worth regressions (Jiao et al., 2004b).

To analyze the sensitivity of association rule mining, a total number of 18×18=324 runs of ARMS are set up by enumerating all combinations of the *min_sup* and *min_conf* values, where both the *min_sup* and *min_conf* values are changed from 0.05 to 0.95 with an increment of 0.05. Using utility data of vibration motors in Jiao et al. (2004a) and process data of vibration motors in Jiao et al. (2003), the result of sensitivity analysis is obtained. As shown in Figure 4-17, the performance measure in Eq. (26) is presented as a normalized comparison. The result of sensitivity analysis suggests that the optimal criteria of association rule mining are given as the support and confidence thresholds of 0.5 and 0.6, respectively.

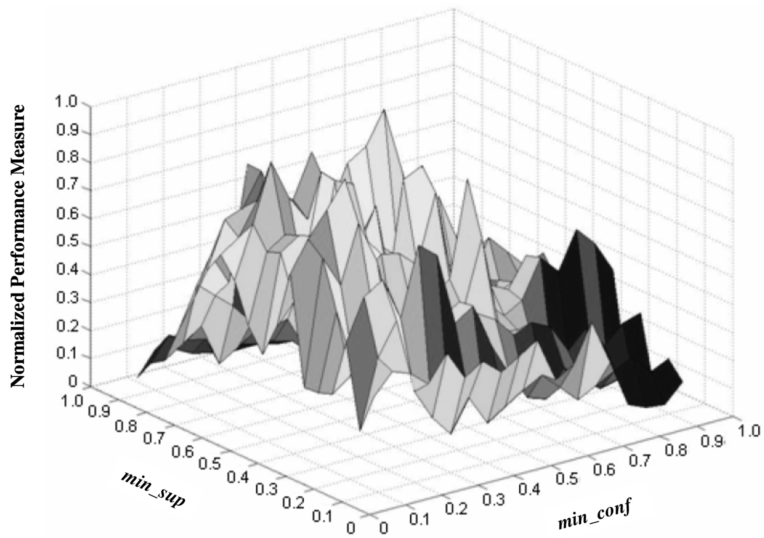


Figure 4-17 Sensitivity analysis of product portfolio identification with respect to minimum support and confidence levels

4.6. Summary

As witnessed in the case study, it is profound to discern CNs from FRs in the respective customer and functional domains. Such a contextual difference in requirement information, as a matter of fact, constitutes the major tradeoffs inherent in the product definition process. While customers concern about the "Alarm" function of a mobile phone, designers have to interpret the implications of these CNs in terms of the functional specification of a vibration motor. During this process, engineering concerns play different roles in analyzing CNs and FRs. In accordance, product portfolio identification should seek for a synergy of these two sets of requirement information so as to achieve the desired "dynamic" functional variety while keeping "stability" in technical variety (Du et al., 2001). Therefore, the ARMS specifies a portfolio in terms of clusters of FRs while bearing on correspondence to CNs. We believe this is more reasonable than most models in market research and requirement management, in which customer groups, market segments, or requirement patterns are all built upon the assumption that CNs and FRs connote the same semantic set of requirements. In this sense, ARMS is more applicable to those consumer products than capital products (industrial products, e.g., power supplies). Consumer products usually involve more explicit interfaces between customers and engineering, whereas capital products involve less explicit customer involvement in engineering. In addition, knowledge recovery by data mining should be more useful for variant designs rather than new designs. Moreover, we advocate the importance of reusing knowledge from past data in order to deliver mass customization within the existing capabilities. In this regard, the portfolio identification has to conform to the product and process platforms that have been installed in the company. So the specification of product offerings in a portfolio indeed represents the functional view of the product and process platforms.

In terms of requirement pattern recognition, association rule mining is advantageous over the traditional method based on decision trees. The key difference between the two techniques lies in that the decision tree method can only produce rules that are mutually exclusive, while association rule mining can produce rules that may not be mutually exclusive (Berson et al., 1999). The reason behind this originates from the way they operate. Association rule mining seeks to go from the bottom up and collect all possible patterns that are of interest, and then use these patterns for some prediction targets. Decision trees, on the other hand, work from a prediction target downward in a manner known as a “greedy” search. They look for the best possible split on the next step. Furthermore, decision trees deal with data records that belong to the same category, whereas association rule mining can handle data records from different itemsets.

Nevertheless, the applicability of ARMS requires intensive collaboration with domain experts and considerations of particular problem contexts. Decisions on the proper similarity threshold and reasonable support and confidence levels may be too complex, and tricky as well, for enterprise managers. In practice, this can be alleviated through iterative interactions between portfolio identification and portfolio evaluation, as what we have done in the sensitivity analysis. Usually, a few scenarios with different settings of these parameters are identified and then input into ARMS. Based on the results, their performances are evaluated against a few pre-defined business objectives. Then the best setup is determined, and the portfolio specification is refined. Hence, portfolio identification and its evaluation are iterative in implementation and thereby should be integrated within a unified framework of product portfolio planning.

While data mining techniques excel in identifying hidden patterns of mapping relationships between CNs and FRs, a practical data mining application is often complex, involving a number of interactive and iterative steps (Han and kamber, 2001). The processing of data throughout the data mining process deserves particular attention for the achievement of good results. This is, however, often neglected and difficult to implement in practice. Pyle (1999) provides a comprehensive coverage of existing data preparation techniques, including discretization, dimensionality reduction, normalization, etc. Treatment of missing values and data cleaning are important exercises for the implementation of data mining. The post-processing of discovered patterns is also important. This may involve interpreting association rules, analyzing the patterns automatically or semi-automatically, or identifying those truly interesting and useful patterns for the user. Also important is to extract target data sets from transaction records based on a thorough understanding of the application domain and the application goals.

As for association rule mining, the support-confidence framework has been the subject of much criticism. The confidence measure does not adequately capture the intuitive and natural semantics of direct associations, in which the associations are obvious (Adamo, 2001). To improve this, Brin et al. (1997) propose an alternative measure, called conviction, to account for the strength of direct associations. In addition, the support-confidence framework tends to favor those rules with dense consequent. As a result, the rule generation process inclines to overstress those rules with a high consequent support. For instance, certain biased rules involving negated attributes are likely to appear in the outcome, making it contain many spurious rules (Aggarwal and Yu, 1998). Towards this end, a number of improvements have been proposed, including improvement-based rule pruning, collective strength, correlated attribute-set enumeration, intensity measure, and so on (Adamo, 2001). Moreover, traditional association rule mining adopts only a single minimum support in rule generation; however, classification data often contains a huge number of rules, which may cause combinatorial exploration. To tackle such an unbalanced data class distribution, Liu et al. (1998) introduce the use of multiple class minimum supports to rule generation by assigning a different minimum support for each class. By incorporating appropriate measures into the association rule mining process, the quality of the rules could be improved dramatically. For example, the Magnum Opus data mining tool employed in this study provides five instruments: coverage, support, strength, lift, and leverage. In the current mining process, we have only used two of them: support and strength. Conjoint use of all these five measures could improve the predictive accuracy of association rule mining substantially (http://www.rulequest.com/MOnew.html). However, the challenge lies in how to apply appropriate measures in accordance with the specific problem context of domain applications.

CHAPTER 5
PRODUCT PORTFOLIO OPTIMIZATION BASED ON HEURISTIC GENETIC ALGORITHM

In this chapter, the product portfolio optimization problem is formulated (see Section 5.1). To leverage both customer and engineering concerns, a maximizing shared surplus model, considering customer preferences, choice probabilities and platform-based product costing, is proposed (see Sections 5.2 and 5.3). A heuristic genetic algorithm procedure is applied to solve the mixed integer combinatorial optimization problem involved in product portfolio optimization (see Section 5.4). Initial findings from a case study of notebook computer portfolio optimization suggest the importance of the research problem, as well as the feasibility and potential of the proposed framework (see Section 5.5). Sensitivity analysis is conducted to evaluate the system performance (see Section 5.6). The chapter concludes with a discussion (see Section 5.7).

5.1. Problem Formulation

This research addresses the product portfolio optimization problem with the goal of maximizing an expected surplus from both customer and engineering perspectives. More specifically, we consider a scenario where a large set of product attributes, $A \equiv \{a_k \mid k = 1,\cdots,K\}$, have been identified based on customer needs (the available method is discussed in Chapter 4, for example, Jiao and Zhang, 2005), given that the firm has the capabilities (both design and production) to produce all these attributes. Each attribute, $\forall a_k \in A$, possesses a few levels, either discrete or continuous, i.e., $A_k^* \equiv \{a_{kl}^* \mid l = 1,\cdots,L_k\}$.

One advantage of using discrete levels is that it does not presume linearity with respect to the continuous variables (Train, 2003).

A set of potential product profiles, $Z \equiv \{\vec{z}_j \mid j = 1,\cdots,J\}$, are generated by choosing one of the levels for certain attributes, subjective to satisfying certain configuration constraints. That is, a product assumes certain attribute levels that correspond to a subset of A. Each product, $\forall \vec{z}_j \in Z$, is defined as a vector of specific attribute levels, i.e., $\vec{z}_j = \left[a^*_{kl_j}\right]_K$, where any $a^*_{kl_j} = \varnothing$ indicates that product $\vec{z}_j$ does not contain attribute a_k; and any $a^*_{kl_j} \neq \varnothing$ represents an element of the set of attribute levels that can be assumed by product $\vec{z}_j$, i.e., $\{a^*_{kl_j}\}_K \in \{A^*_1 \times A^*_2 \times \cdots \times A^*_K\}$.

A product portfolio, Λ, is a set consisting of a few selected product profiles, i.e., $\Lambda \equiv \{\vec{z}_j \mid j = 1,\cdots,J^\dagger\} \subseteq Z$, $\exists J^\dagger \in \{1,\cdots,J\}$, denotes the number of products contained in the product portfolio.

The cost of offering product $\vec{z}_j$ is denoted as $\{C_j\}_J$. The manufacturer must make decisions about which products to offer as well as their respective prices, $\{p_j\}_J$. As for portfolio decisions, the manufacturer must also determine what combinations of attributes and their levels should be introduced, or be discarded from product offerings. This is different from traditional product line design, which involves the selection of products only, yet leaves the sets of attributes and their levels intact, and assumes the products are generated a priori by enumerating all possible attribute levels. In this sense, this research adopts a one-step approach to the optimal product line design problem, which excels in simultaneously optimizing product generation and selection when facing a large number of combinations of attributes and their levels (Steiner and Hruschka, 2002).

There are multiple market segments, $S \equiv \{s_i \mid i = 1, \cdots, I\}$, each containing homogeneous customers, with a size, Q_i. The customer-engineering interaction is embodied in the decisions associated with customers' choices of different products. Various customer preferences on diverse products are represented by respective utilities, $\{U_{ij}\}_{I \cdot J}$. Product demands or market shares, $\{P_{ij}\}_{I \cdot J}$, are described by the probabilities of customers' choosing products, denoted as customer or segment-product pairs, $\{(s_i, \vec{z}_j)\}_{I \cdot J} \in S \times Z$.

Customers choose a product based on the surplus buyer rule (Kaul and Rao, 1995). They have the option of not buying any product (if none produces a positive surplus) or buying competitors' products. Assume that competitors do not respond to the manufacturer's moves, meaning that, in the short run, the competition does not react by introducing new products. This is supported by the findings of Robinson (1988). As a result, competitive reactions appear implicitly in the customer utilities, which are influenced by the attributes and prices of competing products. In addition, assume that neither price nor supply discrimination is allowed. That is, each offered product bears the same price for all segments, and each segment can buy any of the products offered (Yano and Dobson, 1998). Moreover, assume that customers can access complete information regarding the available products and their prices. The growing presence of electronic commerce for business-to-business and business-to-customer sales is also expanding the availability of product and price information.

5.2. Optimization Model

As discussed in Chapter 3, a maximizing shared surplus model is proposed to leverage both the customer and engineering concerns inherent in the product portfolio optimization problem. The objective function is formulated as the following:

$$Maximize \quad E[V] = \sum_{i=1}^{I}\sum_{j=1}^{J}\frac{U_{ij}}{C_j}P_{ij}Q_i y_j \,, \tag{27}$$

where $E[\cdot]$ denotes the expected value of the shared surplus, V , which is defined as the utility per cost, modified by the probabilistic choice model, $\{P_{ij}\}_{I \cdot J}$, and the market size, $\{Q_i\}_I$, C_j indicates the cost of offering product $\bar{z}_j$, and y_j is a binary variable such that $y_j = 1$ if the manufacturer decides to offer product $\bar{z}_j$ and $y_j = 0$ otherwise.

To select the best product portfolio with nearly the same shared surplus, a selection rule is adopted to identify the most balanced product portfolio. According to Li and Azarm (2002), a balanced product portfolio means that all products contribute evenly or nearly evenly to the shared surplus; otherwise, it is an unbalanced product portfolio. In general, a balanced product portfolio is more preferable, as it tends to perform more stably when unexpected changes occur in the market. An unbalanced product portfolio, on the contrary, may suffer significantly when market changes diminish the performance of one or two dominating products in the portfolio. To quantify the extent of a balanced distribution of products' individual contributions to the entire product portfolio, an unbalanced index is defined as the following:

$$\psi = \sqrt{\sum_{j=1}^{J^{\dagger}}\left(\frac{E[V_j]}{E[V]} - \frac{1}{M}\right)^2} \,, \tag{28}$$

where M is the total number of products in a portfolio, $E[V_j]$ is the expected shared surplus of product $\vec{z}_j$, and $E[V]$ is the expected shared surplus of all products, $\{\vec{z}_j\}_{J^\dagger}$. In an absolutely balanced portfolio, the shared surplus of portfolio is evenly distributed among all products, i.e., $\frac{E\left[V_j\right]}{E[V]} \to \frac{1}{M}$, thus $\psi \to 0$. Therefore, the lower the value of the unbalanced index is, the more balanced is the distribution of shared surplus (fitness) among the products, and thus the more desirable is the portfolio.

5.2.1. Conjoint Analysis and Customer Preference

Given a set of attributes and their levels, conjoint analysis starts with a factorial design. To avoid the combinatorial explosion problem if all possible pairings of attribute levels are used, an efficient design is required (Green and Krieger, 1996). The design of experiments technique can be used to select the attribute combinations. The factors of an experimental design are variables that have a few levels. Experiments are performed to study the effects of the factor levels on the response, or dependent variable. In a conjoint study, the factors are the attributes of the potential products, and the response is a rating or ranking of customer preferences. The rows of a design are called runs and correspond to product profiles in a full-profile conjoint study. A special type of fractional-factorial design is the use of an orthogonal array. An orthogonal array helps reduce the number of combinations using efficient designs that are both orthogonal and balanced, and hence optimal. Efficient designs specially tailored to conjoint studies are supported in a number of software packages, such as Sawtooth, SPSS and SAS (Wittink and Cattin, 1989).

When conjoint utilities are measured using a continuous function, either a quadratic function (Pekelman and Sen, 1979) or a vector model of preference, the resulting utility function can be directly applied to the planning model in Eq. (27). Among many preference models used in conjoint analysis, part-worth models are most general and widely used in commercial applications (Wittink and Cattin, 1989). Therefore, this research adopts a linearization of part-worth for the analysis, although any continuous function can be used without loss of generality.

Following the part-worth model, the utility of the i-th segment for the j-th product, U_{ij}, is assumed to be a linear function of the part-worth preferences (utilities) of the attribute levels of product $\vec{z}_j$, i.e.,

$$U_{ij} = \sum_{k=1}^{K} \sum_{l=1}^{L_k} \left(w_{jk} u_{ikl} x_{jkl} + \pi_j \right) + \varepsilon_{ij}, \tag{29}$$

where u_{ikl} is the part-worth utility of segment s_i for the l-th level of attribute a_k (i.e., a_{kl}^*) individually, w_{jk} is the utility weights among attributes, $\{a_k\}_K$, contained in product $\vec{z}_j$, π_j is a constant associated with the derivation of a composite utility from part-worth utilities with respect to product $\vec{z}_j$, ε_{ij} is an error term for each segment-product pair, and x_{jkl} is a binary variable such that $x_{jkl} = 1$ if the l-th level of attribute a_k is contained in product $\vec{z}_j$ and $x_{jkl} = 0$ otherwise.

There are a number of methods available to estimate regression utility weights, $\{w_{jk}\}_{J \cdot K}$, and the constant, $\{\pi_j\}_J$, given a set of observed choice data, including full-profile conjoint analysis, adaptive conjoint analysis, hybrid conjoint analysis, and experimental choice analysis, or choice-based conjoint analysis (see http://www.sawtoothsoftware.com). In addition, a great deal of research in marketing has been devoted to recovering model parameters through latent classes, such as using finite mixtures, hierarchical Bayes methods, the maximum likelihood formulation, and the least squares method (Lilien et al., 1992).

In the above formulation, customer behavior is modeled at the segment level, although one could also assume individual level part-worth utilities without loss of generality. As observed by Wittink and Cattin (1989), market segmentation ranks among the primary purposes of suppliers in conjoint studies. If segmentation issues are of particular interest, individual level part-worth estimations might further be clustered to form market segments (i.e., post hoc segmentation). Moreover, a number of procedures for simultaneously performing market segmentation and calibrating segment-level part-worth utilities in conjoint analysis have been developed in recent years. Such methods for simultaneous segmentation and estimation have been proposed for both the traditional conjoint analysis and the choice-based conjoint analysis (Wedel and Kamakura, 1998).

5.2.2. Choice Model and Product Demand

Conjoint analysis yields a preference model, for example a main-effect part-worth model, which defines the functional relationship between attribute levels of a product and a customer's or a segment's overall utility attached to it. Based on this preference model, customers' choices can be modeled by relating preference (utility) to choice. The traditional deterministic first choice rule of preferences assumes that a customer chooses the product from the choice set according to the highest associated utility with certainty. Neglect of uncertain factors in the first choice rule may lead to suboptimal results at the aggregate market level, as market shares of products with higher utilities across customers or segments tend to be overestimated (Kaul and Rao, 1995).

Probabilistic choice rules can provide more realistic representations of the customer decision making process (Sudharshan et al., 1987). Some probabilistic choice rules can offer flexibility in calibrating actual choice behavior such as the option of mimicking the first choice rule (Kaul and Rao, 1995). In general, there are two types of probabilistic choice rules (Ben-Akiva and Lerman, 1985): the generalized (or powered) Bradley-Terry-Luce share-of-utility rule and the conditional multinomial logit choice rule (MNL). With the assumption of independently and identically distributed error terms, the logit choice rule suggests itself to be a discrete choice model (Ben-Akiva and Lerman, 1985). Discrete choice models are best suited to estimate customer preferences directly from choice data (Green and Krieger 1996) – the case of product portfolio optimization, where customers' choices are directed to the attribute levels that constitute products. Moreover, with discrete choice models, preference estimation and model calibration can be performed simultaneously and tests for statistical inferences about a particular model and its parameters are available (Ben-Akiva and Lerman, 1985). Therefore, this research employs the logit choice rule to model product demands.

Under the MNL model, the choice probability, P_{ij}, that a customer or a segment, $\exists s_i \in S$, chooses a product, $\exists \vec{z}_j \in Z$, with N competing products, is defined as the following:

$$P_{ij} = \frac{e^{\mu U_{ij}}}{\sum_{n=1}^{N} e^{\mu U_{in}}}, \tag{30}$$

where μ is a scaling parameter. As $\mu \to \infty$, the logit behaves like a deterministic model, whereas it becomes a uniform distribution as $\mu \to 0$. Therefore, as with the BTL model, calibration on actual market shares can be carried out subsequently to elaborate preference estimation by post hoc optimization with respect to μ (Train, 2003).

Based on a customer survey, the response rate - how often each product alternative is chosen - can be depicted as a probability density distribution. The demand for a particular product is the summation of the choice frequency of each respondent, $\forall s_i \in S$, adjusted for the ratio of respondent sample size versus the size of the market population (Train, 2003). The accuracy of the demand estimates can be increased by identifying unique customer utility functions per market segment, or class of customers to capture systematic preference variations (Ben-Akiva and Lerman, 1985). Estimates of future demand can also be facilitated using pattern-based or correlation-based forecasting of existing products. Forecasts of economic growth and the estimated change of the socioeconomic and demographic background of the market populations help to refine these estimates (Lilien et al., 1992).

5.2.3. Dealing with Engineering Costs

The premise of existing profit-maximizing approaches is to assume that costs can be estimated, provided that the manufacturer has established an operating cost accounting system (Dobson and Kalish, 1993). As discussed in Chapter 3, cost estimation, however, is deemed to be very difficult, especially at the portfolio optimization phase. The cost advantages in mass customization rest with the achievement of mass production efficiency. Rather than the absolute amount of dollar costs, what is important to justify optimal product offerings is the magnitudes of deviations from existing product and process platforms due to design changes and process variations in relation to product variety. Therefore, Jiao and Tseng (2004) have proposed to model the cost consequences of providing variety based on varying impacts on process capabilities. The process capability index lends itself to an instrument for handling the sunk costs related to product families and shared resources.

To circumvent the difficulties inherent in estimating the accurate cost figures, this research adopts a pragmatic costing approach based on standard time estimation developed by Jiao and Tseng (1999b). The idea is to allocate costs to those established time standards from well-practiced work and time studies, thus relieving the tedious tasks for identifying various cost drivers and cost-related activities. The key is to develop mapping relationships from different attribute levels to their expected consumptions of standard times within legacy process capabilities. These part-worth standard time accounting relationships are built into the product and process platforms (Jiao et al., 2003). Any product configured from available attribute levels is justified based on its expected cycle time. This expected cycle time is accounted by the aggregation of part-worth standard times. The rationale is particularly applicable to portfolio optimization, where “the optimal product profiles are not as sensitive to absolute dollar costs as they are to the relative magnitudes of cost levels” (Choi and DeSarbo, 1994).

The expected cycle time can be used as a performance indicator of variations in process capabilities (Jiao and Tseng, 2004). The characteristic for the cycle time is of ‘the smaller the better’ type. The cycle time demonstrates the distinctions between variables that differ as a result of random error and are often well described by a normal distribution. Hence, the one-side specification limit process capability index, PCI, can be formulated as the following:

$$PCI = \frac{USL^T - \mu^T}{3\sigma^T}, \tag{31}$$

where USL^T, μ^T, and σ^T are the upper specification limit, the mean and the standard deviation of the estimated cycle time, respectively. Variations in the cycle time are characterized by μ^T and σ^T, reflecting the compound effect of multiple products on production in terms of process variations. The USL^T can be determined ex ante based on the worst case analysis of a given process platform, in which standard routings can be reconfigured to accommodate various products derived from the corresponding product platform (Jiao et al., 2003).

The value of PCI falls between $[0,1]$, where a large value suggests the related production process is easy to implement (as it involves little deviation from existing platforms), and a small value a difficult one. As there exist close correlations between cost and cycle time, the PCI can indicate how expensive a product is expected to be if produced within the existing capabilities. Introducing a penalty function, the cost function, C_j, corresponding to product $\vec{z}_j$, can be formulated based on the respective process capability index, PCI_j, that is,

$$C_j = \beta e^{\frac{1}{PCI_j}} = \beta e^{\frac{3\sigma_j^T}{USL^T - \mu_j^T}}, \tag{32}$$

where β is a constant indicating the average dollar cost per variation of process capability, USL^T denotes the upper limit of cycle times for all product variants to be produced within the process platform, μ_j^T and σ_j^T are the mean and the standard deviation of the estimated cycle time for product $\vec{z}_j$, respectively.

The estimated cycle time for product $\vec{z}_j$, (μ_j^T, σ_j^T), is assumed to be a linear function of the part-worth standard times of the attribute levels assumed by product $\vec{z}_j$, modified by the probabilistic choice model, $\{P_{ij}\}_{I \cdot J}$ and the market size, $\{Q_i\}_I$, i.e.,

$$\mu_j^T = \sum_{k=1}^{K} \sum_{l=1}^{L_k} \left(\zeta_{jk} \mu_{kl}^t x_{jkl} + \omega_j \right) \times \sum_{i=1}^{I} P_{ij} Q_i \,, \tag{33a}$$

$$\sigma_j^T = \sqrt{\sum_{k=1}^{K} \sum_{l=1}^{L_k} \left(\sigma_{kl}^t x_{jkl} \right)^2 \times \sum_{i=1}^{I} P_{ij} Q_i} \,, \tag{33b}$$

where ζ_{jk} and ω_j are regression coefficients, x_{jkl} possesses the same meaning as that in Eq. (29), and μ_{kl}^t and σ_{kl}^t are the mean and the standard deviation of the part-worth standard time associated with the l-th level of attribute a_k, respectively.

The meaning of β is consistent with that of the dollar loss per deviation constant widely used in Taguchi's loss functions. It can be determined ex ante based on the analysis of existing product and process platforms. Such a cost function produces a relative measure, instead of actual dollar figures, for evaluating the extent of process variations among multiple products. Modeling the economic latitude of product portfolio optimization through the cycle time performance and the impact on process capabilities can alleviate the difficulties in traditional cost estimation, which is tedious and less accurate.

5.3. Model Development

Surplus-based optimization models assume customers only choose the product with a positive surplus as opposite to the lowest price. Otherwise, the price of each offered product becomes a decision variable, making the problem nonlinear (Yano and Dobson, 1998). To avoid explicitly, nor necessary, modeling of the price, the general practice is to treat the price as a separate attribute that can be chosen from a limited number of values for each product (Nair et al., 1995; Moore et al., 1999). Adding the price as one more attribute, the attribute set becomes $A \equiv \{a_k\}_{K+1}$, where a_{K+1} represents the price possessing a few levels, i.e., $A^*_{K+1} \equiv \{a^*_{(K+1)l} \mid l = 1,\cdots,L_{K+1}\}$. Let $\vec{p} = \left[a^*_{(K+1)1},\cdots,a^*_{(K+1)L_{K+1}}\right]'$ be the vector of feasible price levels. Further let $\vec{x}_{j(K+1)l}$ be a binary vector of length L_{K+1} indicating the presence or absence of the l-th price level with respect to product $\vec{z}_j$. Then $p_j = \vec{p} \otimes \vec{x}_{j(K+1)l}$ suggests the price assigned to product $\vec{z}_j$.

Combining Eqs. (27), (29), (30) and (32), the product portfolio optimization problem can be formulated as a mixed integer program, as below:

$$Maximize \quad E[V] = \sum_{i=1}^{I}\sum_{j=1}^{J}\left(\frac{U_{ij}}{\beta e^{\frac{3\sigma_j^T}{USL^T-\mu_j^T}}}\right)\left(\frac{e^{\mu U_{ij}}}{\sum_{n=1}^{N} e^{\mu U_{in}}}\right) Q_i y_j \, , \tag{34a}$$

$$s.t. \quad U_{ij} = \sum_{k=1}^{K+1}\sum_{l=1}^{L_k}\left(w_{jk}u_{ikl}x_{jkl} + \pi_j\right) + \varepsilon_{ij}\, , \qquad \forall i \in \{1,\cdots,I\}, \forall j \in \{1,\cdots,J\}, \tag{34b}$$

$$\sum_{l=1}^{L_k} x_{jkl} = 1\, , \qquad \forall j \in \{1,\cdots,J\}, \forall k \in \{1,\cdots,K+1\}, \tag{34c}$$

$$\sum_{k=1}^{K+1}\sum_{l=1}^{L_k}\left|x_{jkl} - x_{j'kl}\right| > 0\, , \qquad \forall j, j' \in \{1,\cdots,J\},\ j \neq j', \tag{34d}$$

$$\sum_{j=1}^{J} y_j \leq J^{\dagger}\, , \qquad \forall J^{\dagger} \in \{1,\cdots,J\}, \tag{34e}$$

$$x_{jkl}, y_j \in \{0,1\}, \qquad \forall j \in \{1,\cdots,J\}, \forall k \in \{1,\cdots,K+1\}, \forall l \in \{1,\cdots,L_k\}. \tag{34f}$$

Objective function (34a) is to maximize the expected shared surplus by offering a product portfolio consisting of products, $\{\vec{z}_j\}_J$, to customer segments, $\{s_i\}_I$, each with size Q_i. Market potentials, $\{Q_i\}_I$, can be given exogenously at the outset or estimated through a variety of techniques based on historical data or test markets (Lilien et al., 1992). Constraint (34b) refers to conjoint analysis – ensures that the composite utility of segment s_i for product $\vec{z}_j$ can be constructed from part-worth utilities of individual attribute levels, $\{A_k^*\}_{K+1}$. Constraint (34c) suggests an exclusiveness condition – enforces that exactly one and only one level of each attribute can be chosen for each product. Constraint (34d) denotes a divergence condition – requires that several products to be offered must pairwise differ in at least one attribute level. Constraint (34e) indicates a capacity condition – limits the maximal number of products that can be chosen for each segment. It can be an inequality or equality. In the case of an inequality constraint, $J^\dagger$ is the upper bound on the number of products that the manufacturer wants to introduce to a product portfolio, whereas with an equality constraint, $J^\dagger$ is the exact number of products contained in a product portfolio. Constraint (34f) represents the binary restriction with regard to the decision variables of the optimization problem.

In the mathematical program of Eq.(34), there are two types of decision variables involved, i.e., x_{jkl} and y_j, representing two layers of decision-making in portfolio optimization, respectively. The first layer is the selection of attributes and their levels for different products (i.e., product generation); the second one decides which products to offer (i.e., product selection). Both types of decisions depend on a simultaneous satisfaction of the target segments. The manufacturer's decisions about *what* (i.e., layer I decision-making) and *which* (i.e., layer II decision-making) products to offer to the target segments are implied in various instances of $\{x_{jkl} \mid \forall j,k,l\}$ and $\{y_j \mid \forall j\}$, respectively. As a result, an optimal product portfolio, $\Lambda^{\dagger} \equiv \{\vec{z}_j^{\dagger} \mid j = 1,\cdots,J^{\dagger}\}$ is yielded as a combination of selected products corresponding to $\{y_j \mid \forall j\}$, where each selected product , $\vec{z}_j^{\dagger}$, comprises a few selected attributes and the associated levels corresponding to $\{x_{jkl} \mid \forall j,k,l\}$. The framework and solution procedures for product portfolio optimization are schematically shown in Figure 5-1, where a heuristic genetic algorithm solver is developed to solve the mixed integer optimization problem.

5.4. Heuristic GA-based Solution

Product portfolio optimization has its origins in the fields of optimal product design (Krishnan and Ulrich, 2001), product positioning (Kaul and Rao, 1995) and product line design (Kohli and Sukumar, 1990). All of these problems constitute a type of combinatorial optimization problems due to their purpose of achieving a near-optimal combination of discrete products and/or attribute levels (Nair et al., 1995). In general, combinatorial optimization problems are characterized by a finite number of feasible solutions. Let $E = \{e_1, e_1, \cdots, e_n\}$ be a finite set, Ω a set of feasible solutions defined over E, and $f : \Omega \rightarrow R$ an objective function. A combinatorial optimization problem is to find a solution in Ω whose objective value is minimum or maximum (Nemhauser and Wolsey, 1988). By intuition, finding the near-optimal solution for a finite combinatorial optimization problem could be done by simple enumeration. In practice, however, this technique is often impossible because the number of feasible solutions may be enormous.

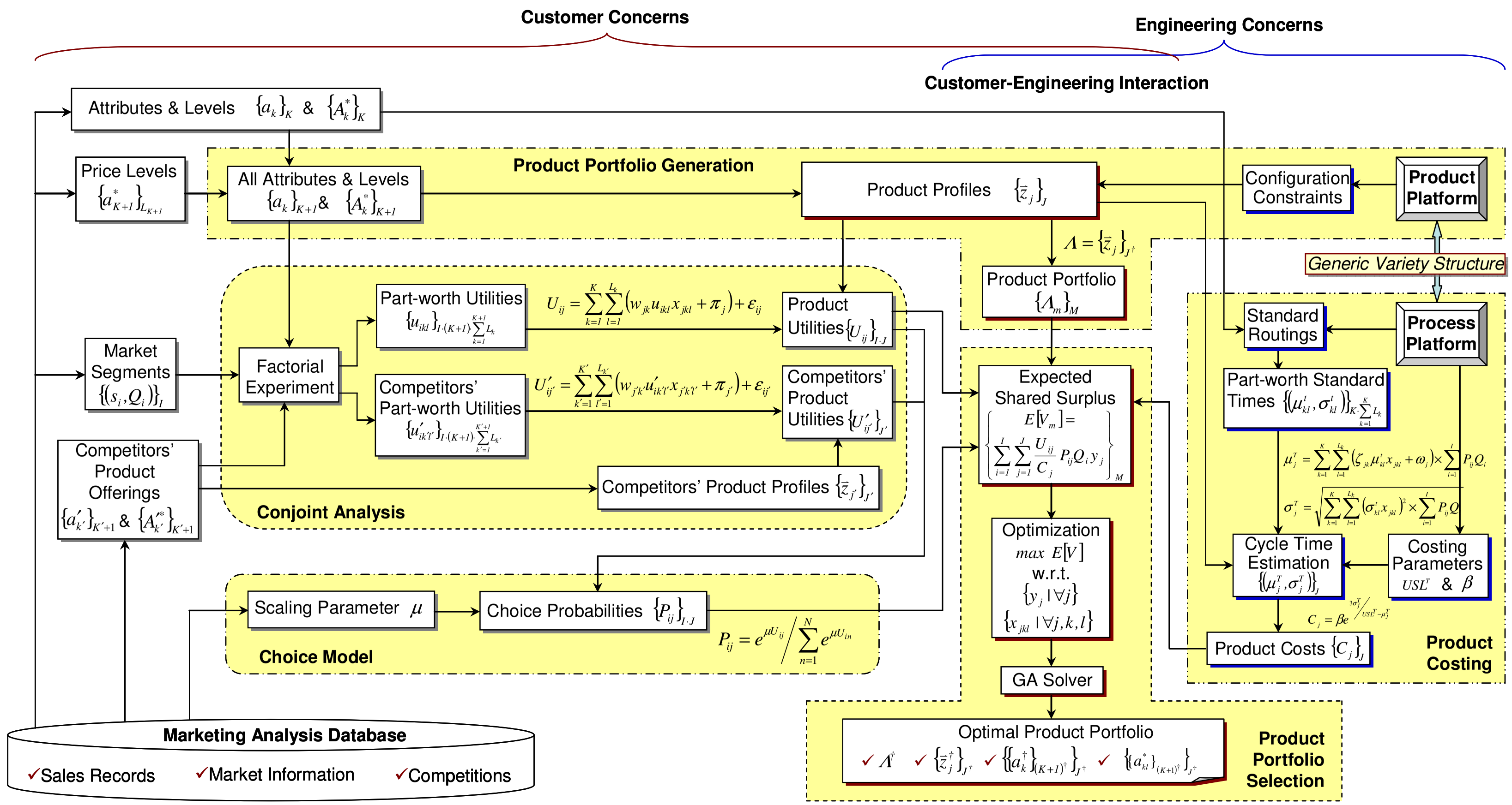


Figure 5-1 Solution schema for product portfolio optimization

Comparing with traditional calculus-based or approximation optimization techniques, genetic algorithms (GA) have been proven to excel in solving combinatorial optimization problems (Steiner and Hruschka, 2002). GA is done from a population of points, rather than a single point (as with branch-and-bound and other techniques), thus increasing the exploratory capability. Objective function information is used directly for evaluation, rather than derivatives used by gradient search techniques. Genetic algorithms evaluate specified candidate solutions completely versus building profiles one attribute at a time. Except for this, GAs work with a direct coding of parameters, rather than the parameters themselves.

Hence, a heuristic GA approach is employed in this research to solve the mixed integer program in Eqs. (34a-f). The focus is to develop an efficient algorithm that is capable of producing acceptable solutions for the combinatorial optimization problem involving a wide variety of configurations of attributes and their levels as well as product profiles in portfolio optimization. In accordance with a generic variety structure inherent in product families (Du et al., 2001), a heuristic GA is formulated as follows.

5.4.1. Generic Encoding

The first step in the implementation of a heuristic GA involves the representation of a problem to be solved with a finite-length string called chromosome. A generic strategy for encoding the portfolio optimization problem is illustrated in Figure 5-2, with an example shown in Figure 5-3. A product portfolio is represented by a chromosome consisting of a string. Each fragment of the chromosome (i.e., substring) represents a product contained in the portfolio. Each element of the string, called a gene, indicates an attribute of the product. The value assumed by a gene, called an allele, represents an index of the attribute level instantiated by an attribute. A portfolio (chromosome) consists of one to many products (fragments of chromosome), exhibiting a type of composition (AND) relationships. Likewise, each product (fragment of chromosome) comprises one to many attributes (genes). Nevertheless, each attribute (gene) can assume one and only one out of many possible attribute levels (alleles), suggesting an exclusive all (XOR) instantiation.

The format of an allele may be a binary, integer, or real value number (Holland, 1992). Hassan et al. (2004) use binary encoding scheme to find optimal product lines with common technology choices. Simptson and D'Souza (2004) adopt real value encoding scheme to design product platform. For portfolio optimization, each attribute (gene) may assume multiple levels (alleles), resulting in a multi-selection problem. Therefore, the integer format is adopted to represent multiple choices among attribute levels. Each gene assumes an integer number corresponding to the index of the attribute level associated with a particular attribute.

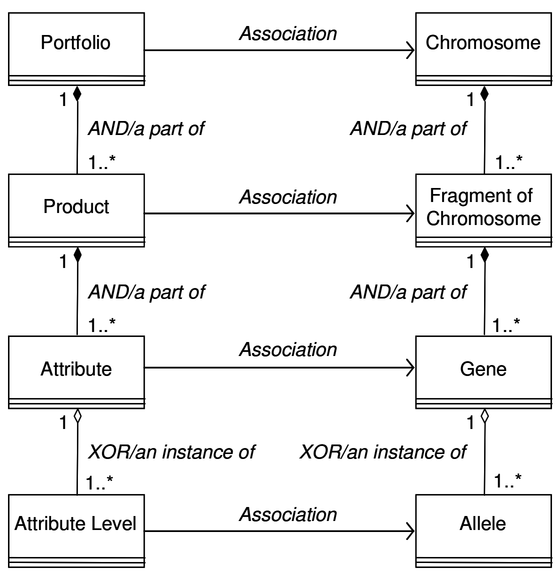


Figure 5-2 Generic encoding for product portfolio

Given $J^{\dagger} \leq J$ products to be selected for a product portfolio, $\Lambda = \{\vec{z}_j\}_{J^{\dagger}}$, and $K+1$ attributes in each product, $\vec{z}_j$, a generic string of the chromosome is defined to be composed of J substrings, with $J - J^{\dagger}$ empty substrings corresponding to those unselected products, and containing a total number of $J \cdot (K+1)$ genes, with each substring consisting of $K+1$ genes.

Further we introduce an allele equal to 0 as the default value for every gene. This indicates that the corresponding attribute is not contained in a product. Then with L_k possible levels for an attribute, a_k, the corresponding gene may assume an allele from the set, $\{0, 1, \cdots, L_k\}$, meaning that a total number of $L_k + 1$ alleles are available for each gene. This corresponds to the fact that an attribute, a_k, may assume a de facto level, that is, $\exists a_{kl}^{*} \in \{\varnothing, a_{k1}^{*}, \cdots, a_{kl}^{*}, \cdots, a_{kL_k}^{*}\}$. If all genes throughout a substring assume $\{0\}_{K+1}$ for the alleles, then it means that the corresponding product is not selected in the portfolio. In this way, a chromosome enables a unified structure, through which various portfolios consisting of different numbers of products can be represented within a generic product portfolio, $\Lambda = \{\vec{z}_j\}_J$. Each individual portfolio can be instantiated from the same generic product portfolio by indirect identification of zero or non-zero alleles for all substrings (Du et al., 2001).

For example, the chromosome shown in Figure 5-3 suggests that product $\vec{z}_J$ is not selected for the portfolio (i.e., $y_J = 0$) as the corresponding substring is totally empty. As far as product $\vec{z}_1$ is concerned (i.e., $y_1 = 1$), the 1st allele assumes a value of 2 indicating that the 1st attribute of the product chooses the 2nd attribute level associated with this attribute (i.e., $x_{112} = 1$). The last ($K+1$) allele of the 1st substring suggests that the price attribute takes on the 3rd price level for product $\vec{z}_1$ (i.e., $x_{1(K+1)3} = 1$). On the other hand, the k-th allele assumes a value of 0, indicating that the k-th attribute is not contained in product $\vec{z}_1$ (i.e., $x_{1kl} = 0$, $\forall l \in \{1, \cdots, L_k\}$).

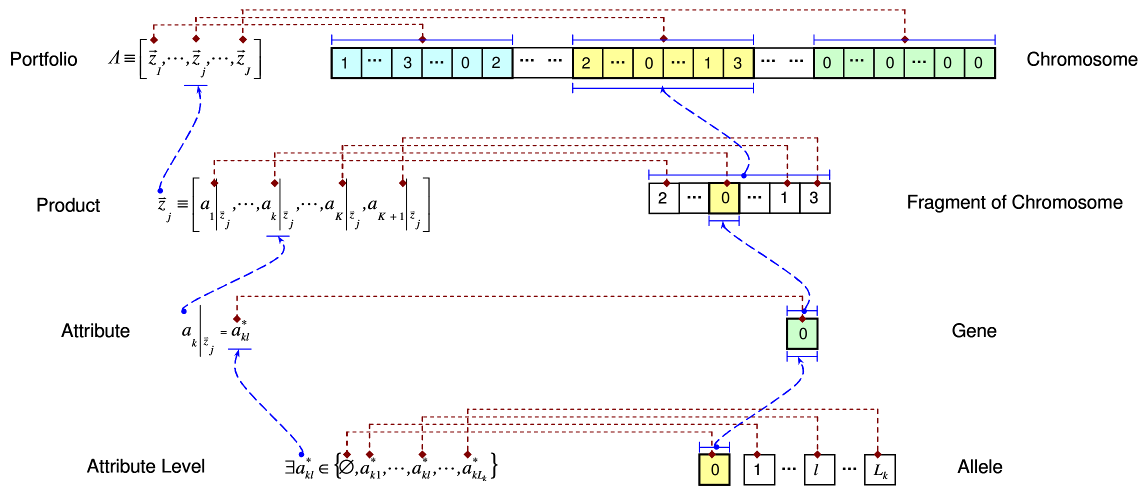


Figure 5-3 An illustration of generic encoding

Following the basic GA procedures (Gen and Cheng, 2000), the product portfolio optimization problem is solved iteratively, as depicted below and also shown in Figure 5-4.

5.4.2. Initialization

Initialization involves generating initial solutions to the problem. The initial solutions can be generated either randomly or using some heuristic methods (Obitko, 2003). Considering the feasibility of product configurations, an initial population of product portfolios of size M, $\{\Lambda_m\}_M$, is determined a priori and accordingly M chromosome strings are encoded, respectively. Each chromosome string is assigned a fitness value in lieu of its expected shared surplus obtained by calculating Eq. (34a).

The population size, M, directly affects the computational efficiency of the GA. A larger population size gives the algorithm a higher chance of success by exploring a larger solution space, but it leads to more calculations. Empirical findings by extensive experimentation have suggested a population size of 100 would produce good solutions for complex problems (Holland, 1992). This research sets a population size of 100 chromosomes.

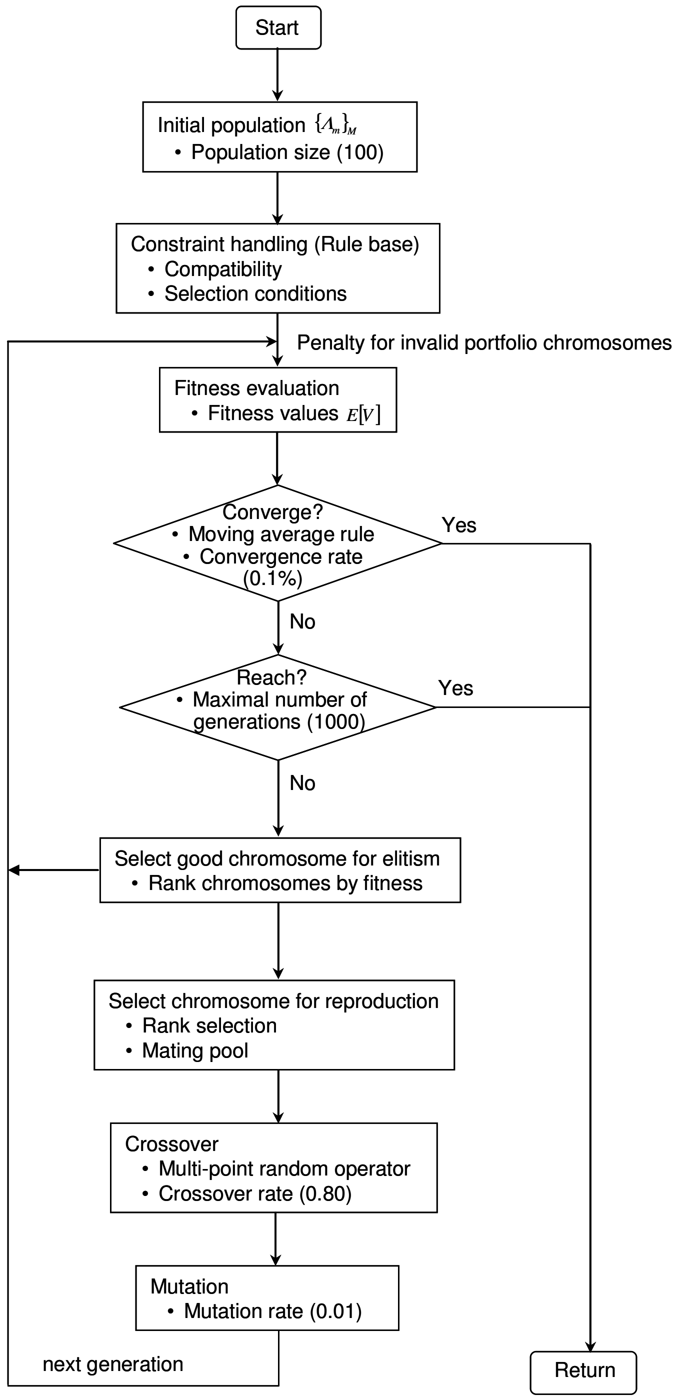


Figure 5-4 Procedure of the heuristic genetic algorithm

5.4.3. Handling of Configuration Constraints

In order to obtain feasible solutions, each chromosome must satisfy certain configuration constraints on product generation from combinations of attribute levels. They constitute two types of constraints: compatibility constraints and selection constraints. Compatibility constraints refer to the restrictions on choices of attribute levels (e.g., size compatible) and are generally described as IF THEN rules (Du et al., 2001; Jiao et al., 2004a). Selection constraints refer to those conjoint, exclusiveness, divergence and capacity conditions as postulated in Eqs. (34b-e).

A number of methods of constraint handling have been reported in the literature, such as the repairing, variable restricting, and modifying generic operator methods (Gen and Cheng, 2000). This research adopts a penalizing strategy. Whenever a new chromosome is generated, a constraint check is conducted with respect to all types of constraints, and those invalid ones are penalized in the population.

Most existing GA implementations incorporate constraint handling into the GA process. This makes GA operations very complex and less efficient. For example, Steiner and Hruschka (2002) have introduced extra exit conditions for crossover and mutation in order to deal with the divergence constraint. This research designs a separate constraint check module as a filter at the outset of the GA process. The constraint rules are generated based on the designers' experience and production capability. The generated rules are stored in a pool. Whenever a new chromosome is produced, it must be checked with the pool. If any genes of the new chromosome are found in the pool, the chromosome is penalized. As a result, only valid chromosomes are kept high fitness, while a standard GA process can be maintained without being intervened by concerning the validity of GA operations or the feasibility of each offspring.

5.4.4. Fitness Function

A fitness function must be used to evaluate the fitness value of each individual chromosome within the population of each generation. Good chromosomes should probably be exposed to more opportunities to be selected as a parent, whereas poor ones may not be selected at all. Within the context of product portfolio optimization, the fitness function used is the expected shared surplus as described in Eq. (34a).

5.4.5. Selection and Reproduction

With the optimization of an expected shared surplus, the fitness values are continuously increasing until a near-optimal solution is found. Once the fitness function is defined and used for the first generation, the GA starts the parent selection and reproduction process. Parent selection is a process that allocates reproductive opportunities among chromosome population. The most popular selection method is the roulette wheel selection. The roulette wheel selection is one probabilistic selection method, that is, a reproduction probability is assigned to each chromosome based on its fitness value. Then the roulette wheel is filled using the respective cumulative probabilities of every chromosome. The areas of the sections on the wheel depend on the fitness values of the associated chromosomes, with fitter chromosomes occupying larger areas in this biased roulette wheel, thus increasing their chances of survival. The roulette wheel selection can be implemented by generating random numbers between 0 and 1 in accordance with the cumulative reproduction probabilities (Obitko, 2003).

The advantage of probabilistic selection is that the better the chromosomes are, the more chances to be selected they have. Thus, those chromosomes with better fitness gain more opportunities to change their good components to reproduce better offspring, but a biased selection sometimes may lead to premature convergence although it enables the convergence of the search (Holland, 1992). Imagine a roulette wheel selection where all the chromosomes in the population are placed, the size of the section in the roulette wheel is proportional to the value of the fitness function of every chromosome - the bigger the value is, the larger the section is. In this case, if one chromosome is dominant in the population, then this dominant chromosome with bigger fitness value will be selected more times. For example, if the best chromosome fitness is 90% of the sum of all fitness values, then the other chromosomes will have very few chances to be selected. Thus, the diversity of the population is destroyed, so as the performance of the global searching capability of genetic algorithm. In this case, this research adopts rank selection to select the appropriate chromosomes for crossover and mutation operations. The rank selection is also a probabilistic selection method. Rank selection ranks the population first, and then every chromosome receives fitness value determined by this ranking. The worst will have a fitness of 1, the second worst 2, etc., and the best will have a fitness of N (number of chromosomes in population). Rank selection decreases the difference between dominant chromosomes and non-dominant ones, thus all the chromosomes have a chance to be selected to keep the diversity of the population.

5.4.6. Crossover

After reproduction, pairs of parent strings in the mating pool are picked randomly, and each pair of strings undergoes crossover with a probability. Crossover requires two individual chromosomes to exchange their genetic compositions. The offspring thus inherits some genes from parents via such operations. While a number of crossover operators are available for specific encoding schemes (Obitko, 2003), this research adopts a multi-point random crossover operator. The idea behind multi-point is that parts of the chromosome that contribute to most of the performance of a particular individual may not necessarily be contained in adjacent substrings. Compared with single-point crossover operator, the disruptive nature of multi-point crossover appears to encourage the exploration of the search space, rather than favoring the convergence to highly fit individuals early in the search, thus making the search more robust.

For the product portfolio optimization problem, the product portfolio comprises several different products which are composed of many attributes. The complexity of the problem results in a long string representing the chromosome. Adopting single-point crossover is inclined to keep most adjacent substrings intact thus resulting in the premature. In this regard, for each substring, single-point crossover operator is adopted to encourage its changing. Thus, the whole chromosome is implemented with a multi-point crossover operation.

Within a generic encoding chromosome, for every substring, one crossover point is randomly located, and the integer string of an offspring is first copied from the first parent from the beginning till the crossover point; and then the rest is added by copying from the second parent from the crossover point to the end. The order of combination is reversed for the other offspring. In regard to the generic chromosome, for each substring, there are $(K-1)$ cutting points, and there are in total $J \cdot (K-1)$ cutting points.

The probability of crossover is characterized by a crossover rate, indicating the percentage of chromosomes in each generation that experience crossover. Crossover aims at producing new chromosomes that possess good elements of old chromosomes. Nonetheless it is also desirable to allow some chromosomes, in particular those good ones, to survive without change in the next generation (namely elitism). Therefore, this research adopts a crossover rate of 0.80. In practice, this value could be selected based on sensitivity analysis of trial examples using crossover rates that range, for example, 0.05-0.95.

5.4.7. Mutation

Mutation is applied to each offspring individually after crossover. It randomly picks a gene within each string with a small probability (referred to as mutation rate) and alters the corresponding attribute level at random. This process enables a small amount of random search, and thus ensuring that the GA search does not quickly converge at a local optimum, but it should not occur very often; otherwise, the GA becomes a pure random search method (Holland, 1992). Empirical findings have suggested a mutation rate of 0.01 as a rule of thumb to obtain good solutions (Gen and Cheng, 2000). While reproduction reduces the diversity of chromosomes in a population, mutation maintains a certain degree of heterogeneity of solutions which is necessary to avoid premature convergence of the GA process (Steiner and Hruschka, 2002).

5.4.8. Termination

The processes of crossover and reproduction are repeated until the population converges or reaches a pre-specified number of generations. The number of generations has direct consequence on the performance of the algorithm. A maximal number can be set ex ante at a large number; however, the algorithm may have found a solution before this number is ever reached. Then extra computations may have to be performed even after the solution has been found. Balakrishnan and Jacob (1996) have shown a moving average rule that can provide a good indication of convergence to a solution. More specifically, the GA process terminates if the average fitness of the best three strings of the current generation has increased by less than a threshold (namely convergence rate) as compared with the average fitness of the best three strings over three immediate previous generations.

To leverage possible problems of termination by either convergence or maximal number of generations alone, this research adopts a two-step stopping rule to incorporate both. A moving average rule is used for the first stopping check. The convergence rate is set at 0.1%. In practice, this value could be determined based on sensitivity analysis of trial examples according to the particular problem context. Then a maximal number of generations is specified as the criterion for the second stopping check. In this case, a number of 1000 is used. Similarly, this value could be determined based on trial runs in line with specific problems under study. These two steps complement each other. If the search is very difficult to converge (for example, in the case of a very tight convergence rate), the second stopping criterion helps avoid running the GA process infinitely. If can converge at the near-optimal solution with a few generations, then there is no need to run as many generations as the maximal number.

Moreover, in each generation the highest fitness value is achieved so far, and its corresponding string is updated and stored. This makes sure that the best product portfolio solution found, not only from the final generation but also over all generations, is returned at convergence. Upon termination, the GA returns the product portfolio with the highest fitness (expected shared surplus) as well as the contained products in terms of specific configurations of attribute levels. All intermediate results of each generation (e.g., product portfolio candidates and their fitness values) and some descriptive statistics (e.g., numbers of crossovers and mutations, average population fitness, population standard deviation and status-quo of product portfolio solution) are recorded in the output report. Thus decision makers can track the progress of the GA or examine other feasible product portfolio solutions that are of high fitness values.

5.5. Case Study

The proposed framework has been applied to the notebook computer portfolio optimization problem for a world-leading computer manufacturing company. The company had conducted extensive market studies and competition analyses and projected the trends of technology development in the business sector concerned. Based on existing technologies, product offerings of notebook computers manifest themselves through various instances of a number of functional attributes. For illustrative simplicity, a set of key attributes and available attribute levels are listed in Table 5-1. Among them, “price” is treated as one of the attributes to be assumed by a product. Every notebook computer is thus described as a viable configuration of available attribute levels.

It is interesting to observe the importance of product portfolio optimization in this case study. Taking the “processor” attribute as an example, existing microelectronics technologies have made it possible to achieve CPU performance ranging from Centrino 1.4 GHz up to Centrino 2.0 GHz. As a matter of fact, one of two existing competitors of the company does offer its products with a very fine portfolio, including Centrino 1.4 GHz, 1.5 GHz, 1.6 GHz, 1.7 GHz, 1.8 GHz, and 2.0 GHz. On the other hand, the other competitor only offers Centrino 1.4 GHz, 1.8 GHz, and 2.0 GHz. It hence becomes imperative to justify the right suite of variety for the company’s product portfolio, regardless of the fact that all these attributes levels are technologically feasible. The question lies in whether or not the granularity of product offerings can leverage the resulting costs and complexity with respect to the company's engineering capabilities.

Table 5-1 List of attributes and their feasible levels for notebook computers

Attribute		Attribute Levels		
a_k	**Description**	a^*_{kl}	**Code**	**Description**
a_1	Processor	a^*_{11}	A1-1	Pentium 2.4 GHz
		a^*_{12}	A1-2	Pentium 2.6 GHz
		a^*_{13}	A1-3	Pentium 2.8 GHz
		a^*_{14}	A1-4	Centrino 1.4 GHz
		a^*_{15}	A1-5	Centrino 1.5 GHz
		a^*_{16}	A1-6	Centrino 1.6 GHz
		a^*_{17}	A1-7	Centrino 1.7 GHz
		a^*_{18}	A1-8	Centrino 1.8 GHz
		a^*_{19}	A1-9	Centrino 2.0 GHz
a_2	Display	a^*_{21}	A2-1	12.1" TFT XGA
		a^*_{22}	A2-2	14.1" TFT SXGA
		a^*_{23}	A2-3	15.4" TFT XGA/UXGA
a_3	Memory	a^*_{31}	A3-1	128 MB DDR SDRAM
		a^*_{32}	A3-2	256 MB DDR SDRAM
		a^*_{33}	A3-3	512 MB DDR SDRAM
		a^*_{34}	A3-4	1 GB DDR SDRAM
a_4	Hard Disk	a^*_{41}	A4-1	40 GB
		a^*_{42}	A4-2	60 GB
		a^*_{43}	A4-3	80 GB
		a^*_{44}	A4-4	120 GB
a_5	Disk Drive	a^*_{51}	A5-1	CD-ROM
		a^*_{52}	A5-2	CD-RW
		a^*_{53}	A5-3	DVD/CD-RW Combo
a_6	Weight	a^*_{61}	A6-1	Low (below 2.0 KG with battery)
		a^*_{62}	A6-2	Moderate (2.0 - 2.8 KG with battery)
		a^*_{63}	A6-3	High (2.8 KG above with battery)
a_7	Battery Life	a^*_{71}	A7-1	Regular (around 6 hours)
		a^*_{72}	A7-2	Long (7.5 hours above)
a_8	Software	a^*_{81}	A8-1	Multimedia package
		a^*_{82}	A8-2	Office package
a_9	Price	a^*_{91}	A9-1	Less than $800
		a^*_{92}	A9-2	$800 - $1.3K
		a^*_{93}	A9-3	$1.3K - $1.8K
		a^*_{94}	A9-4	$1.8K - $2.5K
		a^*_{95}	A9-5	$2.5K above

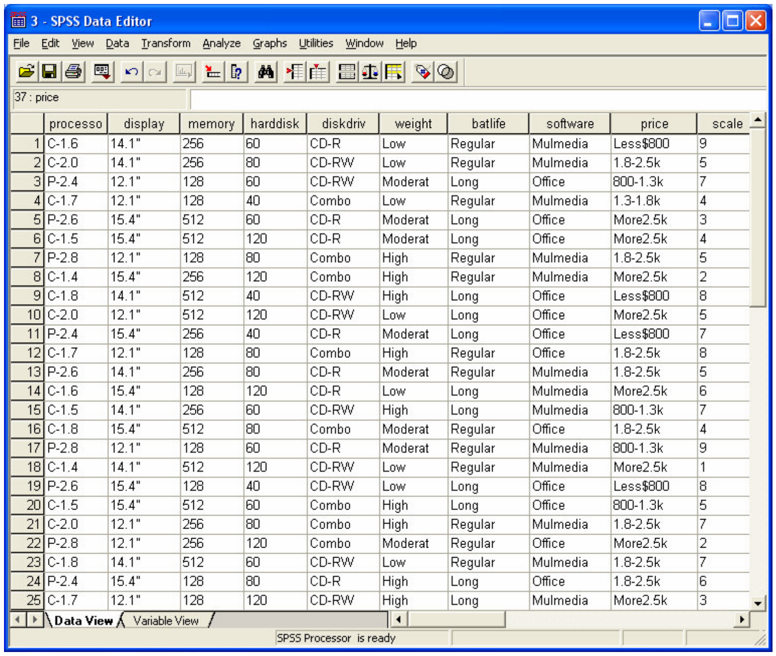
5.5.1. Customer Preference

Conjoint analysis starts with the construction of product profiles. Given all attributes and their possible levels as shown in Table 5-1, a total number of $9 \times 4^2 \times 3^3 \times 2^2 \times 5 = 77760$ possible combinations may be constructed. To overcome such an explosion of configurations with enumeration, orthogonal product profiles are always used in practice (Wittink and Cattin, 1989). Using the Taguchi Orthogonal Array Selector provided in SPSS software (www.spss.com), a total number of 81 orthogonal product profiles are generated, shown as Figure 5-5, comprising 9 factors with each containing 9, 3, 4, 4, 3, 3, 2, 2, and 5 levels, to explore customer preferences. These profiles are explained in Table 5-2, where columns 2-10 indicate the specification of offerings that are involved in the profiles and column 11 collects the preferences given by the customers.

Table 5-2 Response surface experiment design

Conjoint Test										Preference Scale
										least most
Profile	**Processor**	**Display**	**Memory**	**Hard Disk**	**Disk Drive**	**Weight**	**Battery Life**	**Software**	**Price**	1 9
1	C-1.6	14.1”	256	60	CD-R	Low	Regular	Multimedia	<$800	9
2	C-2.0	14.1”	256	80	CD-RW	Low	Regular	Multimedia	$1.8-2.5K	5
3	P-2.4	12.1”	128	60	CD-RW	Moderate	Long	Office	$800-1.3K	7
4	C-1.7	12.1”	128	40	Combo	Low	Regular	Multimedia	$1.3-1.8K	4
…	…	…	…	…	…	…	…	…	…	…
79	C-1.7	15.4”	256	80	CD-R	Moderate	Regular	Multimedia	$1.8-2.5K	3
80	C-1.5	15.4”	1	120	Combo	Low	Regular	Multimedia	$800-1.3K	8
81	C-1.5	14.1”	128	80	CD-RW	High	Regular	Office	$1.3-$1.8K	4

A total number of 30 customers are selected to act as the respondents. Each respondent is asked to evaluate all 81 profiles one by one by giving a mark based on a 9-point scale, where “9” means the customer prefers a product most and “1” least. This results in 30×81 groups of data. Based on this data, clustering analysis is used to find customer segments based on the similarity among customer preferences. Three customer segments are formed: s_1, s_2, and s_3, suggesting home users, regular users, and professional/business users, respectively. These segments, s_1, s_2 and s_3, divide the 30 respondents into three respective groups: (1) customers 1, 2, 7, 8, 11, 12, 14, 15, 24 and 29; (2) customers 3, 4, 5, 9, 10, 13, 17, 19, 20, 23, 26 and 30; and (3) customers 6, 16, 18, 21, 22, 25, 27 and 28.



	processo	display	memory	harddisk	diskdriv	weight	batlife	software	price	scale
1	C-1.6	14.1"	256	60	CD-R	Low	Regular	Mulmedia	Less$800	9
2	C-2.0	14.1"	256	80	CD-RW	Low	Regular	Mulmedia	1.8-2.5k	5
3	P-2.4	12.1"	128	60	CD-RW	Moderat	Long	Office	800-1.3k	7
4	C-1.7	12.1"	128	40	Combo	Low	Regular	Mulmedia	1.3-1.8k	4
5	P-2.6	15.4"	512	60	CD-R	Moderat	Long	Office	More2.5k	3
6	C-1.5	15.4"	512	120	CD-R	Moderat	Long	Office	More2.5k	4
7	P-2.8	12.1"	128	80	Combo	High	Regular	Mulmedia	1.8-2.5k	5
8	C-1.4	15.4"	256	120	Combo	High	Regular	Mulmedia	More2.5k	2
9	C-1.8	14.1"	512	40	CD-RW	High	Long	Office	Less$800	8
10	C-2.0	12.1"	512	120	CD-RW	Low	Long	Office	More2.5k	5
11	P-2.4	15.4"	256	40	CD-R	Moderat	Long	Office	Less$800	7
12	C-1.7	12.1"	128	80	Combo	High	Regular	Office	1.8-2.5k	8
13	P-2.6	14.1"	256	80	CD-R	Moderat	Regular	Mulmedia	1.8-2.5k	5
14	C-1.6	15.4"	128	120	CD-R	Low	Long	Mulmedia	More2.5k	6
15	C-1.5	14.1"	256	60	CD-RW	High	Long	Mulmedia	800-1.3k	7
16	C-1.8	15.4"	512	80	Combo	Moderat	Regular	Office	1.8-2.5k	4
17	P-2.8	12.1"	128	60	CD-R	Moderat	Regular	Mulmedia	800-1.3k	9
18	C-1.4	14.1"	512	120	CD-RW	Low	Regular	Mulmedia	More2.5k	1
19	P-2.6	15.4"	128	40	CD-RW	Low	Long	Office	Less$800	8
20	C-1.5	15.4"	512	60	Combo	High	Long	Office	800-1.3k	5
21	C-2.0	12.1"	256	80	Combo	High	Regular	Mulmedia	1.8-2.5k	7
22	P-2.8	12.1"	256	120	Combo	Moderat	Regular	Office	More2.5k	2
23	C-1.8	14.1"	512	60	CD-RW	Low	Regular	Mulmedia	1.8-2.5k	7
24	P-2.4	15.4"	128	80	CD-R	High	Long	Office	1.8-2.5k	6
25	C-1.7	12.1"	128	120	CD-RW	High	Long	Mulmedia	More2.5k	3

Figure 5-5 Results of orthogonal product profiles

For each respondent in a segment, 81 regression equations are obtained by interpreting his original choice data as a binary instance of each part-worth utility. Each regression corresponds to a product profile and indicates the composition of his original preference in terms of part-worth utilities according to Eq. (29). With these 81 equations, the part-worth utilities for this respondent are derived. Averaging the part-worth utility results of all respondents belonging to the same segment, a segment-level utility is obtained for each attribute level. Columns 2-4 in Table 5-3 show the part-worth utilities of three segments with respect to every attribute level.

Table 5-3 Part-worth utilities and part-worth standard times

Attribute Level	Part-worth Utility (Customer Segment)			Part-worth Standard Time (Assembly & Testing Operations)	
	s_1	s_2	s_3	μ^t (second)	σ^t (second)
A1-1	0.75	0.65	0.62	497	9.5
A1-2	0.77	0.83	0.82	536	11
A1-3	0.81	0.78	1.18	563	12
A1-4	0.74	0.66	0.61	512	10.5
A1-5	0.77	0.86	0.89	556	11.8
A1-6	0.78	0.77	1.16	589	21
A1-7	0.81	0.79	1.18	598	21.1
A1-8	0.83	0.82	1.21	615	22.3
A1-9	0.84	0.85	1.22	637	24
A2-1	1.18	1.05	0.75	739	35
A2-2	1.21	1.47	1.18	819	37
A2-3	1.25	1.49	1.38	836	39
A3-1	1.02	0.5	0.4	659	24.5
A3-2	1.09	0.9	0.65	699	26.5
A3-3	1.12	1.15	0.93	725	32
A3-4	1.14	1.18	1.11	756	36
A4-1	1.33	0.97	0.63	641	26
A4-2	1.38	1.08	0.78	668	28
A4-3	1.52	1.13	1.08	707	29
A4-4	1.56	1.19	1.22	865	40
A5-1	0.86	0.93	0.78	293	4.4
A5-2	0.88	1.11	0.82	321	5.1
A5-3	0.92	1.35	0.83	368	5.5
A6-1	0.7	0.2	0.3	215	3.8
A6-2	0.9	0.7	0.8	256	4.0
A6-3	1.1	0.9	0.9	285	4.1
A7-1	0.7	0.6	0.3	125	1.6
A7-2	0.8	0.9	1.2	458	19.1
A8-1	1.2	1.1	1.2	115	1.55
A8-2	0.5	0.8	1.0	68	0.95
A9-1	0	0	0	N.A.	N.A.
A9-2	-1.75	-0.35	-0.2		
A9-3	-2.25	-0.65	-0.47		
A9-4	-2.75	-2.48	-0.6		
A9-5	-3.5	-3.3	-0.95		

5.5.2. Engineering Cost

Table 5-3 also shows the part-worth standard times for all attribute levels. The company fulfills customer orders through assembly-to-order production while importing all components and parts via global sourcing. The part-worth standard time of each attribute level is established based on work and time studies of the related assembly and testing operations. With assembly-to-order production, the company has identified and established standard routings as basic constructs of its process platform. Based on empirical studies, costing parameters are known as $USL^T = 3\times10^4$ (hours) and $\beta = 460$.

5.5.3. HGA Solution

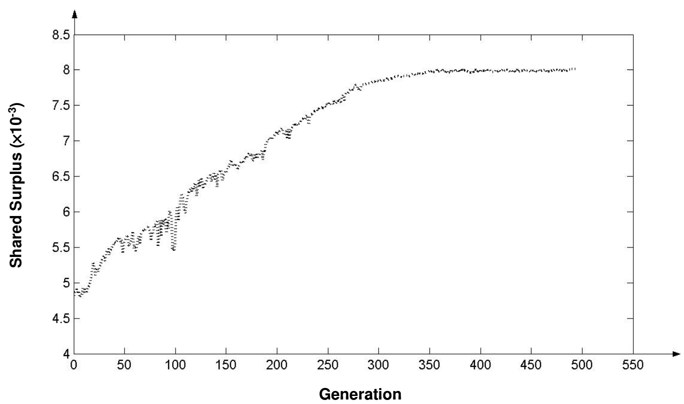
To determine a near-optimal notebook computer portfolio for the target three segments, the GA procedure is applied to search for a maximum of expected shared surplus among all attribute, product and portfolio alternatives. Assume that each portfolio may consist of a maximal number of $J^\dagger = 5$ products. Then a chromosome string comprises $9\times5=45$ genes. Each substring is as long as 9 genes and represents a product that constitutes the portfolio.

Based on the economical analysis, some constraints are generated to ensure the profitability. The constraints are represented by "IF-THEN" rules, that is, "IF $x_1 = 9$, THEN $x_9 \neq 1$; IF $x_1 = 3$, THEN $x_9 \neq 2$; IF $x_3 = 4$, THEN $x_9 \neq 1$; IF $x_4 = 4$, THEN $x_9 \neq 1$". These constraints restrict the customers from buying high performance notebook computer with too low price. For every generation, a population size of $M = 100$ is maintained, meaning that only the top 100 best product portfolios are kept for reproduction.

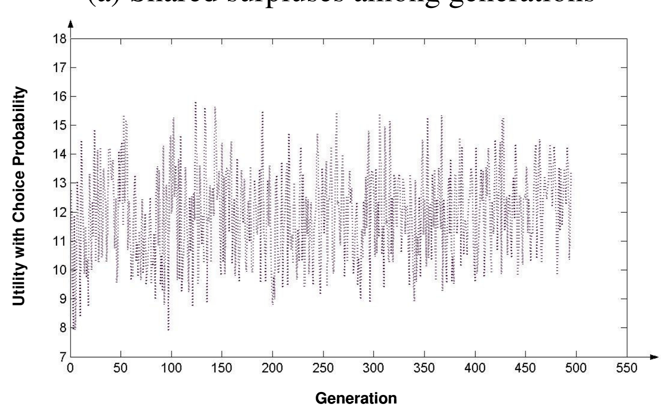
In addition, it is not uncommon that in the notebook computer business most manufacturers directly order components and parts from their suppliers. This means that all the companies possess similar technological capabilities to provide the attributes and levels listed in Table 5-1. In fact, the produceability of those attributes and levels depends on global semiconductor suppliers rather than notebook computer manufacturers themselves. Therefore, we assume that the competitors of the company under this study offer the same product attributes and levels. As a result, the status-quo product alternatives in the current generation are used as the pool of competing products for the choice model in Eq. (29).

5.5.4. Results

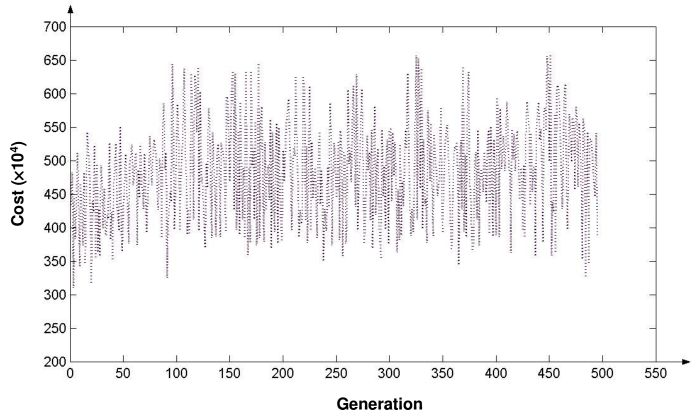
The results of GA solution are presented in Figures 5-6. As shown in Figure 5-6(a), the fitness value keeps improving along the reproduction process generation by generation. Certain local optima (e.g., around 100 generations) are successfully overcome. The saturation period (350-500 generations) is quite short, indicating the GA search is efficient. This proves that the moving average rule is a reasonable convergence measure. It helps avoid such a possible problem that the GA procedure may run unnecessarily as long as 1000 generations. Upon termination at the 495th generation, the GA solver returns the optimal result, which achieves an expected shared surplus of 8.02×10^{-3}, and an unbalanced index of 0.2, as shown in Table 5-4.



(a) Shared surpluses among generations



(b) Utilities with choice probability among generations



(c) Costs among generations

Figure 5-6 Results of GA solution

Table 5-4 Optimal solution of notebook computer portfolio

Product Portfolio $\Lambda^\dagger$	Chromosome $\Lambda^\dagger = [1,2,1,1,1,3,1,1,2;8,3,3,3,3,1,2,0,4;0,0,0,0,0,0,0,0,0;0,0,0,0,0,0,0,0,0;0,0,0,0,0,0,0,0,0]$			
Constituent Products $\{\vec{z}_j^\dagger\}_{J^\dagger}$	Substring $\vec{z}_1^1 = [1,2,1,1,1,3,1,1,2]$		Substring $\vec{z}_2^1 = [8,3,3,3,3,1,2,0,4]$	
Attributes $\{a_k^\dagger\}_{(K+1)^\dagger}$ Attribute Levels $\{a_{kl}^*\}_{(K+1)^\dagger}$	$a_k^\dagger$	a_{kl}^*	$a_k^\dagger$	a_{kl}^*
	Processor	Pentium 2.4 GHz	Processor	Centrino 1.8 GHz
	Display	14.1" TFT SXGA	Display	15.4" TFT XGA/UXGA
	Memory	128 MB DDR SDRAM	Memory	512 MB DDR SDRAM
	Hard Disk	40 GB	Hard Disk	80 GB
	Disk Drive	CD-ROM	Disk Drive	DVD/CD-RW Combo
	Weight	High (2.8 KG above)	Weight	Low (below 2.0 KG)
	Battery Life	Regular (around 6 hours)	Battery Life	Long (7.5 hours above)
	Software	Multimedia package	∅	Nil
	Price	\$800 - \$1.3K	Price	\$1.8K - \$2.5K
Expected Shared Surplus $E[V^\dagger]$	8.02×10^{-3}			
Unbalance Index $\psi^\dagger$	0.2			

As shown in Table 5-4, the optimal product portfolio consists of two products, $\vec{z}_1^1$ and $\vec{z}_2^1$. From the specifications of attribute levels, we can see that they basically represent the low-end and high-end notebook computers, respectively. With such a two-product portfolio, all home, regular and professional/business users can be served with an optimistic expectation of maximizing the shared surplus. While low-end notebook computer $\vec{z}_1^1$ includes all available attributes, high-end notebook computer $\vec{z}_2^1$ does not contain the "software" attribute. This may manifest the fact that most professionals prefer to install software authorized by their business organizations for the purpose of, for example, systems maintenance and technical support.

5.5.5. Performance Evaluation

Figure 5-6(b) compares the results of utility with choice probability, $\sum_{i=1}^{3}\sum_{j=1}^{5}\left(U_{ij}P_{ij}\right)y_j$, among generations. It is interesting to observe that the distribution of utility with choice probability does not tally with that of the fitness shown in Figure 5-6(a). The optimal solution (i.e., the last generation) does not produce the best utility performance. On the other hand, a number of high utility achievements do not correspond to high fitness. Likewise, as shown in Figure 5-6(c), the distribution of cost performance among generations disorders the pattern of fitness distribution shown in Figure 5-6(a). This may be explained by the fact that high utility achievement is usually accompanied with high incurred costs. Therefore, the shared surplus is a more reasonable fitness measure to leverage both customer and engineering concerns than either utility or cost alone.

Figure 5-7 compares the achievements, in terms of the normalized shared surplus, cost, and utility with choice probability, of the top 20 product portfolios in the 495th generation that returns the optimal solution. It is interesting to see that the peak of utility achievement (portfolio #6) does not contribute to producing the best fitness as its cost is estimated to be high. On the other hand, the minimum cost (portfolio #5) does not mean the best achievement of shared surplus as its utility performance is low. The best portfolio (#1) results from a good balance between both utility and cost performances.

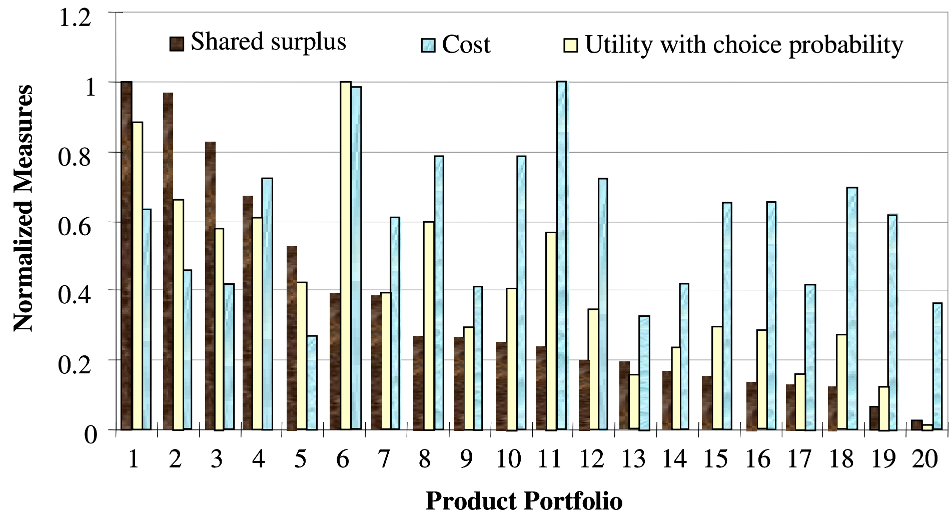


Figure 5-7 Performance comparison of top 20 product portfolio population in the 495th generation

Figure 5-8 shows the performance of individual constituent products in terms of the unbalanced index for the top 20 portfolios in the 495th generation. It is noted that the top five portfolios all contain a moderate number of products (2-3), whereas those portfolios consisting of more products (e.g., portfolios #11, 16 and 19) seldom produce very good performance. This exactly illustrates the granularity tradeoff issue in product portfolio planning. In fact, too many products introduced in a portfolio may even bring about competition among themselves. On the other hand, none of the top 20 portfolios contains only one product. In practice, a single-product portfolio is not a desired case either, as it facilitates a limited coverage of diverse customer segments. Figure 5-8 shows that three product portfolios are outstanding with respect to their shared surplus (portfolio # 1, 2, 3) with their normalized shared surplus of 1, 0.94, and 0.83 respectively. Among these portfolios, portfolio 1 is the best with respect to its unbalanced index with an unbalanced score of 0.2, and thus it is selected as the final choice.

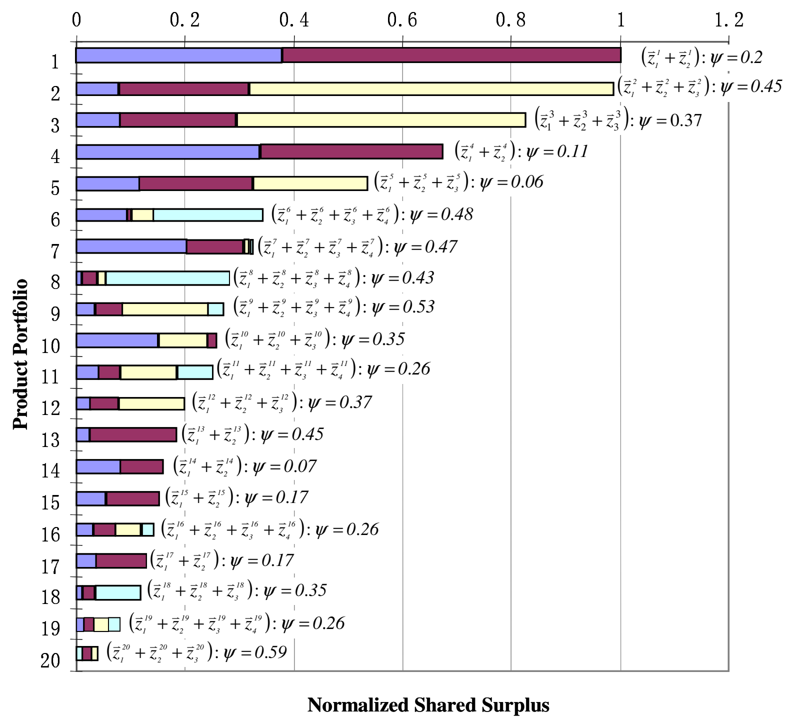


Figure 5-8 Comparison of constituent products for top 20 product portfolios produced in the 495th generation

5.6. Sensitivity Analysis

It is very important to maintain population diversity during the GA searching process. Low diversity may cause "inbreeding", thus weakening the exploratory capability (Laumanns, et al., 2002). Many parameters can influence the population diversity. For example, an excessively high crossover rate will cause the solution to converge quickly before the optimum is found. On the other hand, a low crossover rate decreases the population diversity and results in a long computation time. The mutation rate also influences the GA performance, as it determines the frequency of random search. Generally, a very low mutation rate is recommended to avoid that the GA process becomes a pure random search, which impairs the search capability of the GA. The population size may be the most distinct factor influencing the population diversity. For a complex problem, a large population size is preferred to ensure exploration in a large search space. In this section, the performance of the heuristic GA is evaluated by means of sensitivity analysis. Based on varying parameter values, such as the population size, the crossover and mutation rates, the heuristic GA performance is examined with respect to different problem sizes.

5.6.1. Problem Size

In accordance with different parameter values required for varying problem sizes, three cases are constructed to represent three different problem sizes for notebook computer portfolio specification. The first case represents a simple problem size, where three attributes are selected, including processor, memory, and weight that are of 9, 4, and 3 levels, respectively. The second case corresponds to a moderate problem size, consisting of six attributes, i.e., processor, memory, weight, hard disk, display, and battery life, which assume 9, 4, 3, 4, 3, and 2 attributes levels, respectively. The third case stands for a very complex problem size, in which all nine attributes and their possible levels are considered. Table 5-5 lists all three scenarios.

5.6.2. Experiment Design

The proper parameter values for the population size, the crossover and mutation rates are recommended through sensitivity analysis. To setup the experiments, 4 values are considered for population size, namely 20, 50, 80, and 100. Likewise 3 values of crossover rate (0.6, 0.7, 0.8) and 3 values of mutation rate (0.005, 0.01, 0.03) are used. Therefore, sensitivity analysis experiment is constructed based on a $4\times3\times3$ full design. For more complex analysis, where more values are involved, other experimental design methods, such as orthogonal design and factorial design, can be employed. The values of these parameters are selected based on the rule-of-thumb from most GA applications - a crossover rate of at least 0.6 and a very low mutation rate.

5.6.3. Parameter Selection

The full design generates 36 scenarios. For each scenario, the GA is run 10 times to collect the mean of its fitness values. Thus, the parameter values with respect to each problem size are recommended on the basis of 360 test runs. The average degree of approximation (Ave_App) associated with GA solutions is adopted as the performance indicator of each problem type. The best GA parameter values are recommended as shown in Table 5-5.

As illustrated in Table 5-5, a larger population size is required for a complex problem in order to maintain population diversity. A population of diverse products is necessary to guarantee thorough exploration of the search space so as to achieve a high degree of average approximation. The crossover rate (p_c) of 0.8 is recommended to encourage more chromosomes to exchange their promising parts and to generate the offspring with better performance. It also demonstrates the tendency that a higher crossover rate leads to better approximation. For complex problems, a higher mutation rate is recommended to avoid the search's falling into local optimum. For the simple problem type, a lower mutation rate is recommended so that the search does not become a pure random search.

Table 5-5 Parameter selection with respect to different problem sizes

Problem Type	Number of Attributes	Problem Size	Parameter Value and Performance						Number of Runs
			Population		Crossover		Mutation		
			Size	Ave_App	P_c	Ave_App	P_m	Ave_App	
Simple	3	15	20	97.3%	0.6	95.6%	0.005	96.2%	360
Moderate	6	30	50	96.7%	0.7	96.3%	0.01	95.8%	360
Complex	9	45	100	94.2%	0.8	97.1%	0.01	94.1%	360

Figure 5-9 shows the average degree of approximation with respect to each population size based on an interval of 20 within the range [20, 160] for the complex problem type. The crossover and mutation rates are set to be 0.8 and 0.01, respectively. As illustrated in Figure 5-9, too large a population size (160) may contribute to the improvement of performance to only a modest extent.

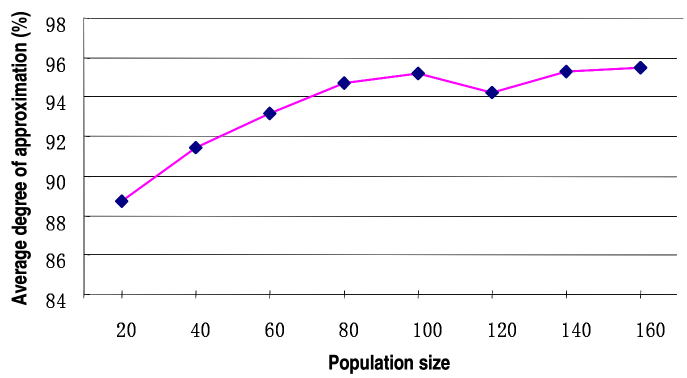


Figure 5-9 Performance with respect to different population sizes

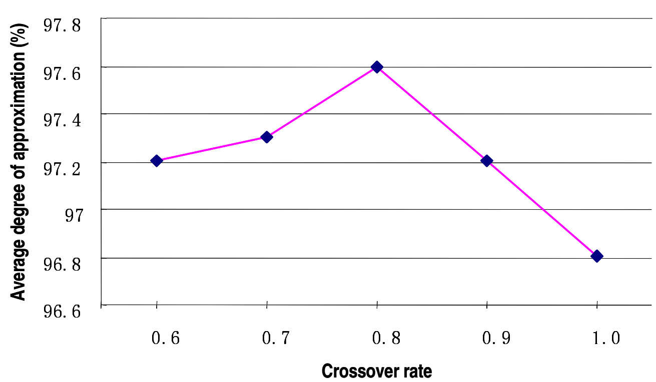


Figure 5-10 Performance with respect to different crossover rate values

Figure 5-10 shows the average degree of approximation for varying crossover rate based on an interval of 0.1 within the range [0.6, 1.0] for the complex problem type. The population size and mutation rate are set to be 100 and 0.01, respectively. It suggests that too large a crossover rate may decrease the performance. This is consistent with previous findings from

GA applications, that is, a large crossover rate may cause too many chromosomes to change, thus leading to premature.

5.7. Summary

Differing from the conventional product line design problem, product portfolio optimization must not only optimize a mix of products but also in the meantime optimize the configurations of individual products in terms of specific attributes. This chapter proposes a maximizing shared surplus model to examine the combined effects of multiple product offerings on both customer preferences and engineering costs. The model allows product portfolios to be constructed directly from part-worth utility and cost.

A heuristic genetic algorithm is developed and applied to solve the combinatorial optimization problem involved in product portfolio optimization. The study indicates that the GA works efficiently in searching for optimal product portfolio solutions. Although the model is used to solve a seller's problem of introducing a new product portfolio with the objective of maximal shared surplus, the proposed framework could easily be adjusted to handle such complex problems as maximizing share-of-choices and extending an existing product portfolio by allowing for already existing items to be owned by the seller. This is supported by the flexibility of the GA procedure that merely uses objective function information, and therefore is capable of accommodating different fitness criteria without any substantial modification of the algorithm.

As demonstrated in the case study, the strength of GA lies in the ability to carry out repeated runs without major changes of parameter values or defining different initial populations, thus improving the chance of finding an optimal or at least a near optimal solution. It is also possible to insert solutions obtained from other techniques into the initial population. Hence, rather than generating all of the members of the initial population at random, the GA can use a prior knowledge about potential optima to arrange the initial population or improve on an existing solution that can perform as a kind of lower bound or benchmark for GA performance.

CHAPTER 6
APPLICATIONS TO CUSTOMER-ENGINEERING INTERACTION

Product portfolio planning lends itself to the discovery of the underlying coupling and interrelationships among various requirements with regard to product performance, as well as the combined effects of multiple product offerings on both customer satisfaction and engineering implications. Thus, it provides insights for applications involving customer-engineering interaction. For example, customers may be supported to make decisions with more engineering concerns. On the other hand, engineering design could be enhanced by capturing more customer satisfaction. In this chapter, three applications are demonstrated: (1) customer decision-making (see Section 6.1); (2) affective design (see Section 6.2); and (3) product family configuration design (see Section 6.3). For each application, the related background knowledge, system architecture, implementation procedure, and validation are described in detail.

6.1. Customer Decision-Making

With the advent of customer-driven marketing, it has been envisioned that e-commerce will emerge as a primary style of manufacturing in the coming decade and beyond (Economist, 2001). The capabilities of e-commerce enable the customer's involvement in design, manufacturing, and service, thus making it possible for product/service providers to interact directly with customers to capture their requirements. A number of online product customization systems have been launched recently (for example, Dell.com, Idtown.com, and

Cannondale.com). These systems support providers to respond to a high variety of requirements and orders by customizing the offerings anticipating the customer requirements.

However, many online customization systems encounter difficulties when dealing with the support for customers' finding the valuable products that match their heterogeneous needs, namely, the personalization problem. It is not uncommon that searching for information or buying complex products (e.g., digital products) via the Internet are always frustrating (Francisco et al., 2005). As in the World Wide Web, the available products and the corresponding amount of electronic information lead to the problem of information overload. Online customers have to access all of the information in order to find what they most prefer. Without face-to-face advice, customers always have difficulties in making tradeoffs among numerous competing products on the Internet. For example, as in real purchase decisions, buyers cannot get all of the best features at the lowest price. In some cases, for specific products, especially for digital products, professional knowledge is always required for evaluation. It is difficult for non-experts to compare products' performances. For example, online customers may be frustrated by the information of digital camera products because they do not know how each feature or its parameters can influence picture quality.

Recommendation systems are traditionally used in e-commence sites to solve the personalization problem by guiding customers to find products they would like to purchase (Yong et al., 2005). A number of recommendation systems have been proposed for different businesses (for example, Group-Lens recommendation system and Ringo). Most of them are either homogeneous (i.e., content-based filtering) or heterogeneous (i.e., collaborative filtering) product recommendation systems (Yuan and Cheng, 2004); however, both of the two paradigms yielded few promising results. The content-based filtering (CBF) approach recommends products to target customers according to the preferences of their neighbors (Hill et al., 1995); however, it is often inhibitive to estimate the preference similarities between various customers. For example, similar preferences may be defined as the preferences of customers who have similar ratings of items (Yoon and Jae, 2004). It is difficult to obtain accurate customer ratings of products especially when special knowledge is needed for rating. The collaborative filtering (CF) approach, on the other hand, recommends products to target customers based on their past preferences (Basu et al., 1998). When facing new customers, this type of recommendation systems cannot recommend a new product as no historical preference records are available (Avery and Zeckhauser, 1997). Nevertheless, both approaches require customers to express their requirements according to system pre-defined formats (e.g., product ratings or customer profiles), and thus real customer requirement information may be distorted.

Due to the drawbacks of traditional approaches, a new paradigm is preferred to advise proper products by capturing accurate individual requirement information (Cheung et al., 2003). As individual customer requirements are heterogeneous, an open environment is required to allow customers to express their diverse requirements completely to their liking. On the other hand, to avoid the difficulties involved in preference estimation, it is preferred to establish such models that allow the prediction of product labels according to customer requirements directly. As a result, the main difficulties involving in establishing recommendation systems for personalization in B2C e-commerce applications can be summarized into two categories. First, customers always use their natural languages to express what they need. Their requirements are normally qualitative and tend to be imprecise and ambiguous due to their linguistic origins. Synonyms are expected to express the same requirements. Further, numerous words that contribute nothing to information retrieval are always found. Second, classification methods have been proven to be an effective means to predict future data objects for which the class label is unknown. Many efficient methods, such as decision trees, regression models, etc., have been developed to identify the relationships between the objects and class labels; however, these methods only excel in classifying the structured data - the object data are organized into a fixed set of attributes or dimensions. Therefore, commonly used relational data-oriented classification methods cannot be adopted to classify customer requirements which are organized into a set of text-based documents.

As discussed in Chapter 4, association rule mining lends itself to the discovery of useful patterns associated with requirement analysis enacted among customers and excels in dealing with semi-structured data. This section presents an associative classification-based recommendation system for personalization in B2C e-commerce applications. A set of associated, frequently occurring text patterns (classifiers) are built by applying an association rule learning method to a training set of requirement text documents. These classifiers are used to predict the product labels for new customer requirements and distinguish one label from others. Thus, products are recommended to customers according to the inner established model that anticipates specific customer needs.

6.1.1. Problem Formulation

By semantic analysis, customer requirements can be described as a set of phrases, $P \equiv \{p_1, p_2, \cdots, p_I\}$. Let $C \equiv \{c_1, c_2, \cdots, c_J\}$ be a set of class labels, each representing a specific product. Suppose there are sales records for S customers and all the sales records comprise a transaction database, T. Every transaction record, $t_s \mid \forall s = 1, 2, ..., S$, comprises the customer requirement record and the record of the product he/she purchased. For each customer $s \mid \forall s = 1, 2, ..., S$ in the transaction database, the corresponding requirement record is described as a set of phrases, P^{t_s}, where $P^{t_s} \subseteq P$. The corresponding product record indicates a class label, $C(t_s)$, where $C(t_s) = c_j \mid \exists\, j \in [1, ..., J]$, showing which product he/she purchased. Thus, transaction records can be summarized as $\langle P, C \rangle$ pairs with the form of $t_s = \{P^{t_s}, C(t_s)\} \mid \forall s = 1, 2, ..., S$. Suppose there are K new object customers for whom the class labels are unknown. The object customers comprise an object database, O, where $O = \{o_k\}_K$. For each customer $k \mid \forall k = 1, 2, ..., K$ in the object database, the requirement record is described as a set of phrases, P^{o_k}, where $P^{o_k} \subseteq P$ and $P^{o_k} \neq P^{t_s} \mid \forall s = 1, 2, ..., S$. Thus, the recommendation problem based on customer requirements is noted as $P^{o_k} \Rightarrow c_j \mid \exists\, j \in [1, ..., J]$, where an association rule, $\Rightarrow$, indicates an inference from the customer requirements (P^{o_k}) to the class label ($c_j \mid \exists\, j \in [1, ..., J]$).

6.1.2. Framework and Methodology: Recommendation System

Based on an associative classification method, an inference system can be constructed for recommendation problems. The system comprises four consecutive stages: (1) the requirement preprocessing module, (2) associative classifier generation module, (3) classification module, and (4) system performance validation module. First, historical requirement data is selected and transformed into proper phrase data sets. Data mining procedure then starts to search for a set of associated, frequently occurring phrase patterns (classifiers). The generated classifiers are pruned by which only those classifiers with good quality are kept for recommendations. When new requirement information comes, the system identifies the corresponding class labels using multiple classifiers. Finally, the performance of the whole system is validated to evaluate how accurately the system will give good recommendations.

6.1.2.1. Requirement preprocessing module

Customer requirements are usually expressed by natural language where many common words occur which contribute nothing to information retrieval. For example, the words “a”, “the”, “of”, “for”, etc., are irrelevant for information even though they may appear frequently. These common words should be filtered out. On the other hand, a group of different words may share the same word stem. To reduce variations in words and increase the scope of searches, these words should be transformed into their canonical forms. In this regard, a stemming algorithm (Porter, 1980) and a common stopword list in English (Fox, 1992) are adopted to reduce the dimensions of the text documents and improve the efficiency of the classifier extraction.

Customer requirements may bear the same semantic meaning even though they are represented by different expressions (Carbonell, 1992). To generalize the requirement information, semantic analysis is adopted. In this research, four thesaurus collections are used to match the requirements. Each collection is composed of several sub-collections, each containing a set of synonyms. The four thesaurus collections are represented as $N=\{N_1,N_2,...\}$, $V=\{V_1,V_2,...\}$, $ADJ=\{ADJ_1,ADJ_2,...\}$, $ADV=\{ADV_1,ADV_2,...\}$, for nouns, verbs, adjectives, and adverbs, respectively. Several semantic rules are represented as IF-THEN rule formats and stored in the semantic rule database to indicate the inference relationship between requirements and a set of predefined phrases, $P\equiv\{p_1,p_2,\cdots,p_I\}$.

Suppose after stopwords removal and stemming, a particular customer requirement is transformed into a word set, $Y \equiv \{y_1, y_2, y_3\}$, and the semantic meaning of such a requirement is represented as IF-THEN rule formats as the following,

IF $y_1 \in V_2$, $y_2 \in ADJ_1$, and $y_3 \in N_3$

THEN the semantic meaning of Y is associated with p_2.

After preprocessing, customer requirements are represented as a set of phrases that are used in the following procedures to generate the classifiers.

6.1.2.2. Associative classifier generation module

As the association rule learning method excels in finding the complex relationships among a huge number of semi- or non-structured items, it is adopted here to generate the classifiers. The general form of associative classifiers is given as the following,

$$\alpha_1 \wedge \alpha_2 \cdots \wedge \alpha_z \cdots \wedge \alpha_Z \Rightarrow \beta \quad [Support = s\%;\ Confidence = c\%], \tag{35}$$

where $\forall\, z = 1, 2, ..., Z$, $\alpha_z = p_i \mid \exists\, i \in [1,...,I]$, $\beta = c_j \mid \exists\, j \in [1,...,J]$; and for any two elements in the precedence, α_x and α_y, where $x, y \in [1,...,Z]$ and $x \neq y$, $\alpha_x \cap \alpha_y = \Phi$, the meanings of the association rule in Eq. (35), as well as $s\%$ and $c\%$, are the same as those discussed in Chapter 4.

6.1.2.3. Classification module

(1) Classifier pruning. For the classifiers generated by association rule learning, one important problem is that the number of the classifiers can be very large. Excessive classifiers extend the time to identify the class labels for given requirement information. Besides, noisy and redundant information impairs the classification quality. To enable the

timely and accurate responses, this research applies CBA-CB algorithm (Liu et al., 1998) to produce the best classifiers out of the whole set of rules.

The driving idea of CBA-CB algorithm is that only those rules that are more general and hold high confidence levels are necessary for the classification task. The unnecessary rules should be pruned by database coverage. Thus, a small number of rules are kept for efficient recommendations. The principles of CBA-CB algorithm are as follows,

(1) Given two rules, $r_x \Rightarrow c_j \mid \exists j \in [1, ..., J]$ and $r_y \Rightarrow c_j \mid \exists j \in [1, ..., J]$, the first rule is more general than the second one if $r_x \subseteq r_y$; and

(2) Given two rules, r_x and r_y, r_x has a higher precedence than r_y, namely $r_x \succ r_y$, if (a) the confidence of r_x ($con(r_x)$) is greater than that of r_y; or (b) $con(r_x) = con(r_y)$, but the support of r_x ($\sup(r_x)$) is greater than that of r_y; or (c) $con(r_x) = con(r_y)$ and $\sup(r_x) = \sup(r_y)$, but r_x is generated earlier than r_y.

Suppose M rules are generated by the classifier generation module and comprise a rule set, R, where $R \equiv \{r_1, r_2, ..., r_M\}$. Each rule, $r_m \mid \forall m = 1,2,...,M$, is pruned according to the first principle. After pruning, general rules are selected and stored in a pruned rule set, R_P. We rank all the rules in R_P in a descendent order according to the second principle and record the ranked rules in a rule set, R_D. For each rule in R_D, $r_m \mid \forall r_m \in R_D$, the phrases involving in its precedent part comprise a set, PR^{r_m}, where $PR^{r_m} \subseteq P$. Let $C(r_m)$ present the class label associated with r_m, where $C(r_m) = c_j \mid \exists j \in [1,..., J]$. We set a cover-count zero for each transaction record, $t_s \mid \forall s = 1,2,..., S$, namely $CC(t_s) = 0$. With respect to each rule,

$r_m \mid \forall r_m \in R_D$, next search all the records in the transaction database. For any record, $t_s \mid \exists s \in [1,...,S]$, if it satisfies the condition, $PR^{r_m} \subseteq P^{t_s}$, it is selected. All the selected transaction records comprise a new set, $T^{'}$, where $T^{'} \subseteq T$. The cover-count is increased by one for all the transaction records in set $T^{'}$. For any rule, $r_m \mid \forall r_m \in R_D$, if it satisfies the condition, $C(r_m) = C(t_s) \mid \exists s \in [1,...,S]$, where $t_s \in T^{'}$, it is put into the filtered rule set, R_F. Then we delete the corresponding rule in set R_D and empty set $T^{'}$. Finally, we delete record $t_s \mid \exists s \in [1,...,S]$ in the transaction database that satisfies the condition, $CC(t_s) \geq \delta \mid \exists s \in [1,...,S]$, where δ is a threshold for cover-count.

(2) Classification based on multiple classifiers. By CBA-CB algorithm, the pruned rules in set R_F are the most significant and finally selected as classifiers to predict the class labels for new requirement information. Suppose the object customers comprise an object database, O, where $O = \{o_k\}_K$. For customer $k \mid \exists k \in [1,...,K]$ in the object database, the requirement record is described as a set of phrases, P^{o_k}, where $P^{o_k} \subseteq P$ and $P^{o_k} \neq P^{t_s} \mid \forall s = 1,2,...,S$. We select rules from set R_F which satisfy the condition, $PR^{r_m} \subseteq P^{o_k} \mid \forall r_m, r_m \in R_F$, and put the selected rules in the classifier rule set, R_C. All the rules in set R_C are grouped based on their associated class labels. Suppose N groups are generated, where $G \equiv \{g_1, g_2, ..., g_N\}$, and each group, $g_n \mid \forall n = 1,2,...,N$, associates with a class label, namely $C(g_n) = c_j \mid \exists j \in [1,...,J]$. Thus, the classification based on multiple classifiers can be formulated as the follows,

$$P^{o_k} \Rightarrow C(g_n) \mid \forall n = 1,2,...,N, \tag{36}$$

$$s.t. \quad \sum_{m} con(r_m) \geq \psi \mid \forall r_m, r_m \in g_n, \tag{37}$$

where P^{o_k} represents the requirement information of customer $k \mid \exists k \in [1,...,K]$ in the object database; $C(g_n)$ means the class label associated with $n-th$ group $\mid \exists n \in [1,...,N]$; $con(r_m)$ is the confidence of rule $r_m \mid \forall r_m, r_m \in R_F$; and ψ is the threshold. Equations (36) and (37) indicate that for the rules selected as classifiers, their associated class labels are selected as recommended ones only if their accumulative confidence satisfies the particular threshold. This enables multiple class labels to be identified based on strong patterns thus adapting to the recommendation problems where multiple recommendations are preferred by allowing customers to make comparisons among a small set of similar products.

6.1.2.4. System performance validation module

To evaluate how accurately the proposed recommendation system assigns class labels according to future customer requirements, this research applies the accuracy measurement (Han and Kamber, 2001) to validate the system performance. A test set is used to measure the recommendation accuracy. Suppose the test set, $\overline{T}$, comprises S records, where $\overline{T} = \{\overline{t_s}\}_S$. For each record in set $\overline{T}$, $\overline{t_s} = \{P^{\overline{t_s}}, C(\overline{t_s})\} \mid \forall s = 1, 2, ..., S$, the associated class labels assigned by the classification module comprise a set $C^{\overline{t_s}}$.

Then the recommendation accuracy is computed using the following,

$$a = \sum_{s=1}^{S} v_s / S, \tag{38}$$

$$s.t. \quad v_s = \begin{cases} 1 & \text{if } C(\overline{t_s}) \in C^{\overline{t_s}} \mid \forall \overline{t_s} \in \overline{T} \\ 0 & \text{otherwise} \end{cases}, \quad \forall s = 1,2,...,S, \tag{39}$$

where a indicates the recommendation accuracy, namely the percentage of the transactions in the test set that are correctly classified; $v_s \mid \forall s = 1,2,\ldots,S$ is a binary variable such that $v_s = 1$ if transaction $\overline{t_s} \mid \forall s = 1,2,\ldots,S$ is correctly classified, and 0 otherwise.

6.1.3. System Analysis and Design

To enable the application of the associative classification-based recommendation system to personalization in B2C e-commerce, this research implements the proposed recommendation system in an Internet programming environment. Figure 6-1 illustrates the function model of the associative classification-based recommendation system. The model comprises five functions. First, customer requirements in the transaction database are extracted and transformed into a set of predefined phrases. The requirement transformation is implemented by the requirement preprocessing function where the stemming algorithm, stopwords removal methodology, and semantic analysis are integrated to process the natural requirements. Allowing the transformed transaction records, the second function, classifier generation function, creates a set of classifiers using the Apriori algorithm. The classifier pruning function then implements the pruning work to remove the noisy and redundant information and the refined rules are stored in the classifier rule database. To validate the system performance, the testing function uses the generated classifiers to assign the class labels for a test set where the class labels are already known. If the performance is validated, the classifiers stored in the classifier rule database are used for future classification tasks. Finally, when new customer requirements occur, they are first processed by the requirement preprocessing function and transformed into the corresponding phrases. Then the classification function searches all the classifiers that satisfy the transformed requirements, thus assigning the corresponding class labels.

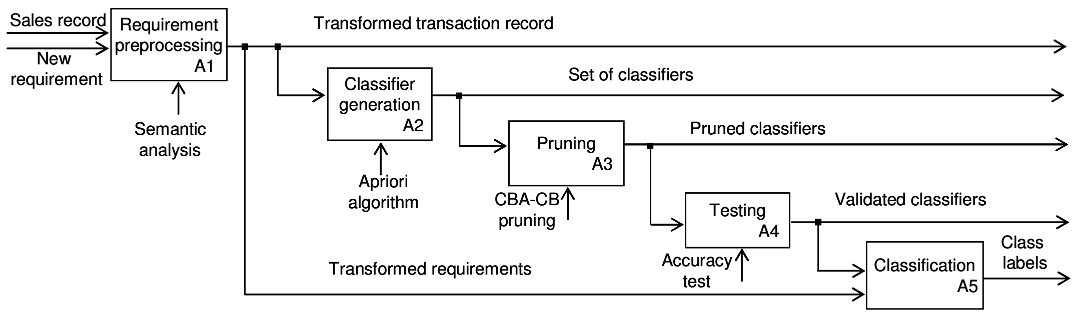


Figure 6-1 Function model of the associative classification-based recommendation system

6.1.4. Web-based Architecture and Implementation

The design, development, and database access for the associative classification-based recommendation system in an Internet environment can be illustrated by the three-tier architecture (Huang and Mak, 2000) as shown in Figure 6-2.

The first tier includes the application clients, namely the customers shopping online. The application clients are involved in the recommendation system only when they are connected with the Web server. There are two types of middle tiers: (1) the Web server and (2) the application server. The Web sites are created by the Web server for applications. The application server is a piece of software to execute the computation activities. In this research, as the pruned classifiers have been generated and stored in the classifier rule database via complex offline work, the Web server and the application server are deployed on the same computer to handle the simplified online tasks. The third tier is the database server to manage the relevant data and rules. In this research, the database is deployed on a computer separate from the Web server.

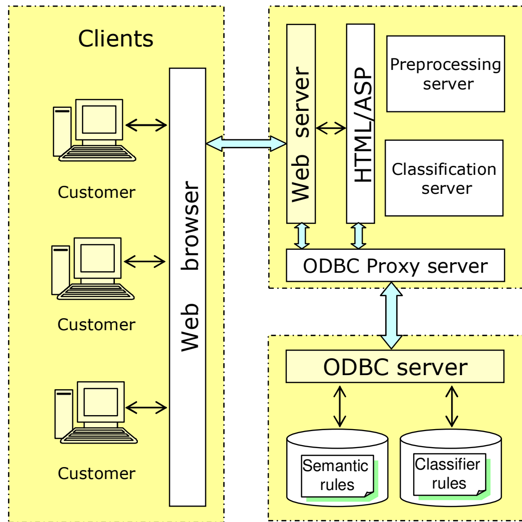


Figure 6-2 Three-tier architecture of the associative classification-based recommendation system in an Internet environment

(1) Client. The clients can be HTML pages, attached components to HTML pages, and programs that can be downloaded from a Web site and then installed, configured and executed on the client machine. Clients and servers communicate with each other through HTTP by exchanging HTML files. For the proposed recommendation system, on the client side, customers, who search and buy products via the Internet, are allowed to log in the Web page and submit their requirement information expressed by their languages to the server and ask for recommendations for the most valuable product alternatives via HTML file format.

(2) Application Server. The application server deals with the computation tasks. It receives client-side requests and information and then processes the data. For the proposed system, the application server comprises two individual servers: the requirement preprocessing server, and the classification server. The requirement preprocessing server deals with the requirement information preprocessing task. When new customer requirement information comes, the requirement preprocessing server searches the semantic rule database

where a set of semantic rules are stored as "IF-THEN" rule formats to indicate the relationships between phrases and natural requirement information. All the rules that satisfy the condition of the requirements are triggered, and the corresponding action clauses (phrases) are identified to match the original requirement information. The classification server assigns the class labels for specific requirement information. Via a lot of offline work, such as classifier generation, pruning, and testing, a set of classifiers are built and stored in the classifier rule database for future classification task. Allowing the new transformed customer requirements, the server finds all the classifiers that satisfy the conditions of such requirements by searching the classifier rule database. The proper class labels are then recommended to match the new requirements based on the identified multiple classifiers.

(3) Database Server. For the conventional approach, if there are changes, the Web designers have to manually adjust all the related categories to reflect the changes. With the database, the Web designers are only required to update those tables containing the related categories without altering the interface. The database server is deployed to manage the data and rules. There are two rule databases in the proposed system, namely, the semantic rule database and the classifier rule database. All of the rules in both of the two databases are described as "IF-THEN" rule formats to represent the relationships between original requirement information and phrases, as well as transformed requirement information and class labels, respectively. To enable the Web application work with rule databases, ODBC, which is a system-level interface communicating with the database, is necessary. ODBC provides a common set of application interfaces (API) to communicate with the database using SQL and Access. For each application server, ODBC is adopted to work with other databases thus integrating diverse applications.

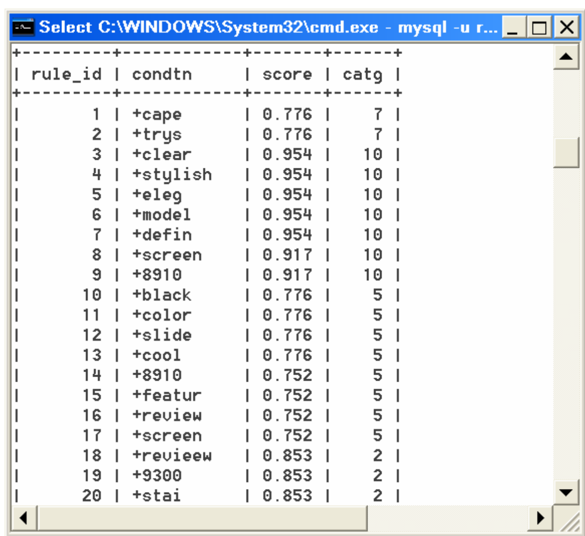
To implement the associative classification-based recommendation system in an Internet environment, this research applies Active Server Pages (ASP) to create the dynamic and interactive personalized pages for the Web site. ASP is a language-independent server-side scripting technology. The most two common scripting languages, VBScript and Jscript, are supported by ASP. ASP provides an open, compile-free application environment in which HTML, scripts, and reusable ActiveX server components can be combined to create dynamic and powerful Web-based solutions. In addition, ASP runs on the server; thus, most browsers will be supported to gain the entire contents of the Web pages to ensure the accessibility to the clients. Further, ASP allows the connection to a database using Active Data Objects (ADO). Data can be simply displayed from an ODBC-compliant database and formatted. To allow the rules to be queried and managed efficiently, this research deploys the Web database query run on Microsoft's Internet Information Server (IIS). All of the rules in the two databases, the semantic rule database and the classifier rule database, are represented as "IF-THEN" (condition-action) formats. Given the query request, the query processor searches for a set of rules whose conditions (IF) satisfy the request. The actions (THEN) of the fired rules are then triggered.

6.1.5. Prototype System and Evaluation

The prototype of the proposed associative classification-based recommendation system has been constructed for mobile phone B2C e-commerce application. Based on the historical sales records, transaction database is established comprising 50 transaction records, as shown in Table 6-1, where customer requirements are described as a set of phrases and the corresponding class labels indicate the mobile phone that has been purchased. Allowing the transaction database, the classifiers are identified. After pruning, the pruned classifiers are used to assign the class labels for future customer requirements. Figure 6-3 shows part of the classifiers where the second column indicates the customer requirements represented as phrases, the third column shows the confidence levels, and the fourth column represents diverse mobile phone products.

Table 6-1 Transaction database records

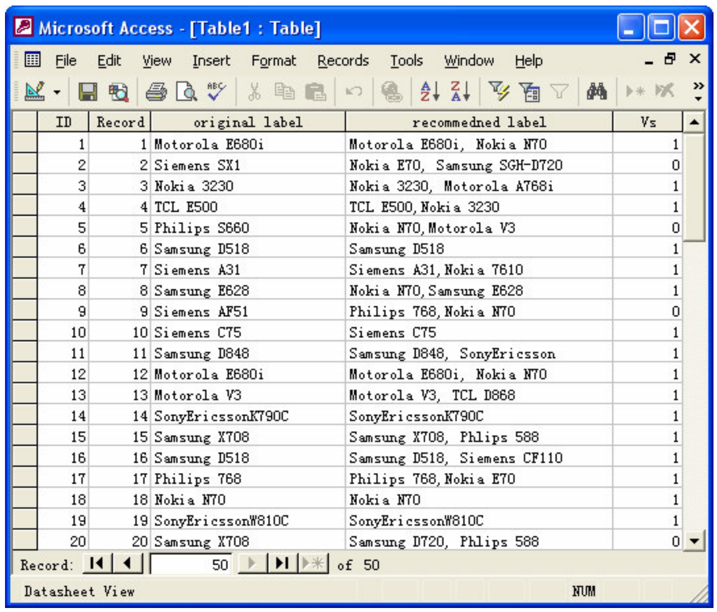
Record t_s	**Requirement Phrases** $P \equiv \{p_1, p_2, \cdots, p_I\}$	**Product Class Label** $C \equiv \{c_1, c_2, \cdots, c_J\}$
t_1	More functions, smallest, lightest, camera, large buttons	Samsung SGH-D730
t_2	Larger screen, office function, usb connectivity, picture transfer	Panasonic X800
t_3	Scheduler, e-mail, news, voice communication	Nokia 9300
…	…	…
t_{48}	Little functions, cheap, high voice quality	Motorola C117
t_{49}	Stylish, elegant, black, video, friendly keypad, small, camera	Nokia 7610
t_{50}	Stylish, elegant, black, light, colorful screen	Nokia 8910



rule_id	condtn	score	catg
1	+cape	0.776	7
2	+trys	0.776	7
3	+clear	0.954	10
4	+stylish	0.954	10
5	+eleg	0.954	10
6	+model	0.954	10
7	+defin	0.954	10
8	+screen	0.917	10
9	+8910	0.917	10
10	+black	0.776	5
11	+color	0.776	5
12	+slide	0.776	5
13	+cool	0.776	5
14	+8910	0.752	5
15	+featur	0.752	5
16	+review	0.752	5
17	+screen	0.752	5
18	+revieew	0.853	2
19	+9300	0.853	2
20	+stai	0.853	2

Figure 6-3 Classifier rule database

The system performance has been validated using 50 test records, shown in Figure 6-4. As explained in Table 6-2, the second column indicates the mobile phone product that has been purchased in each test record, and the third column lists the mobile phone products that are recommended by the associative classification-based recommendation system using classifiers. The value of v_s shown in the fourth column is either one or zero to indicate whether the recommendation system advises the correct products. In this case, 42 test records are correctly recommended, resulting in the recommendation accuracy of 84%. After validation, the generated classifiers are stored in the classifier rule database for further retrieval, update, and query.



ID	Record	original label	recommedned label	Vs
1	1	Motorola E680i	Motorola E680i, Nokia N70	1
2	2	Siemens SX1	Nokia E70, Samsung SGH-D720	0
3	3	Nokia 3230	Nokia 3230, Motorola A768i	1
4	4	TCL E500	TCL E500, Nokia 3230	1
5	5	Philips S660	Nokia N70, Motorola V3	0
6	6	Samsung D518	Samsung D518	1
7	7	Siemens A31	Siemens A31, Nokia 7610	1
8	8	Samsung E628	Nokia N70, Samsung E628	1
9	9	Siemens AF51	Philips 768, Nokia N70	0
10	10	Siemens C75	Siemens C75	1
11	11	Samsung D848	Samsung D848, SonyEricsson	1
12	12	Motorola E680i	Motorola E680i, Nokia N70	1
13	13	Motorola V3	Motorola V3, TCL D868	1
14	14	SonyEricssonK790C	SonyEricssonK790C	1
15	15	Samsung X708	Samsung X708, Phlips 588	1
16	16	Samsung D518	Samsung D518, Siemens CF110	1
17	17	Philips 768	Philips 768, Nokia E70	1
18	18	Nokia N70	Nokia N70	1
19	19	SonyEricssonW810C	SonyEricssonW810C	1
20	20	Samsung X708	Samsung D720, Phlips 588	0

Figure 6-4 Validation records

Table 6-2 System performance results

Record $\bar{t}_s$	**Original Class Label** $C(\bar{t}_s)$	**Recommended Class Labels** $C^{\bar{t}_s}$	v_s
$\bar{t}_1$	Motorola E680i	Motorola E680i, Nokia N70	1
$\bar{t}_2$	Siemens SX1	Nokia E70, Samsung SGH-D720	0
$\bar{t}_3$	Nokia 3230	Nokia 3230, Motorola V3, Motorola A768i	1
…	…	…	…
$\bar{t}_{49}$	Samsung X478	Samsung X478, Nokia 1110, Samsung E568	1
$\bar{t}_{50}$	Samsung E628	Samsung E628, Nokia 9300	1
$\sum_{s=1}^{S} v_s$	42		
a	84%		

Online customers are now allowed to search and buy mobile phone products via the Internet. The dynamic Web pages produced by ASP are not affected by the type of browser the online customer is using. Thus, online customers can access the recommendation system through the Internet more conveniently. Connecting with the database through the ODBC proxy server, data and rules can be easily retrieved to support the application server to process the classification tasks. The most valuable product alternatives are then identified and represented to the online customers via HTML file format. Figure 6-5 shows the recommendation results for two different customer requirements. For example, Figure 6-5(a) indicates that the customer wants to buy a mobile phone that has a scheduler and can send email. The recommendation result is Nokia 9300 shown in Figure 6-5(b), which is deemed as the most valuable mobile phone for the corresponding requirements supported by the associative classification-based recommendation system. Figure 6-5(c) and 6-5(d) show another customer requirement and the corresponding results, where the customer asks for the mobile phone that is stylish, elegant, and in black color, and Nokia 7610 and 8910 are recommended for the customer's further comparison. Supported by the associative classification-based recommendation system, customers are able to find the mobile phone products online that accord with their requirements mostly among numerous available mobile phones. The multiple recommendations also allow customers to make further comparisons among a reduced product set online. This helps the information overload problem, thus improving the efficiency and effectiveness of B2C e-commerce.

Figure 6-5(a) The first customer requirement for mobile phone product

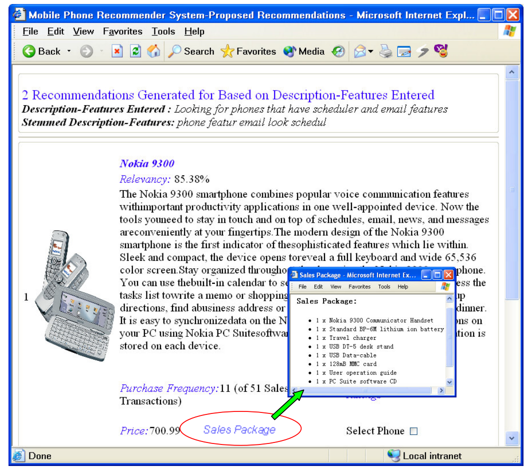


Figure 6-5(b) Recommendation result with respect to the first customer requirement

Figure 6-5(c) The second customer requirement for mobile phone product

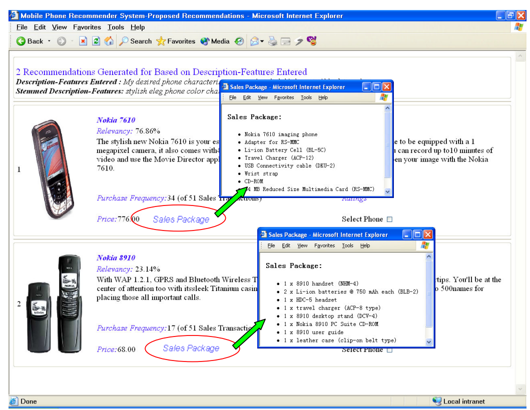


Figure 6-5(d) Recommendation result with respect to the second customer requirement

Figure 6-5 Customer requirements and the recommendation results

6.2. Affective Design

In today's competitive environment, satisfying customer needs has become a great concern of almost every company (Cross, 2000). While there are various customer needs, the functional and affective needs have been recognized to be of primary importance for customer satisfaction (Khalid, 2001). In particular, mass customization and personalization are increasingly accepted as an important instrument for firms to gain competitive advantages (Tseng and Piller, 2003). Moreover, with the development of global markets and modern technologies, it is likely that many similar products will be functionally equivalent. Customers may find it is difficult to distinguish and choose among many product offerings (Huffman and Kahn, 1998). Design for performance (e.g., functional design) and design for usability (e.g., ergonomic design) no longer empower a competitive edge because product technologies turn to be mature, or competitors can quickly catch up (Khalid and Helander, 2004). In this regard, it is imperative to design products by engaging customers' emotions or attention so as to differentiate products from each other.

When designing products, customers' affective needs must be considered (Jordan, 2000). Affect is said to be a customer's psychological response to the perceptual design details (e.g., styling) of the product (Demirbilek and Sener, 2003). Affect is a basis for the formation of human values and human judgment. For this reason it might be argued that models of product design that do not consider affect are essentially weakened (Helander and Tham, 2003). Until recently, the affective aspects of designing and design cognition have been substantially absent from formal theories of design (Helander et al., 2001). Affective design is the inclusion or representation of affect (e.g., emotions, subjective impressions, visual perceptions, etc.) in design processes (Khalid, 2004). Many research issues are implied, including, for example, (1) how to measure and analyze human reactions to affective design; and (2) how to assess the corresponding affective design features. In the end, it is necessary to develop theories and predictive models for affective design.

The main challenge for affective design is to grasp the customers' affective needs accurately and subsequently to design products that match these needs. In most cases, it is very hard to capture the customers' affective needs due to their linguistic origins. Since subjective impressions are difficult to translate into verbal descriptions, affective needs are relatively short-lasting emotional states and tend to be imprecise and ambiguous (Helander and Khalid, 2005). Sometimes, without any technical experience, the customers do not know what they really want until their preferences are violated. In practice, customers, marketing folks and designers employ different sets of context to express their understanding of affect information. Differences in semantics and terminology impair the coherence of transferring affective needs effectively from customers to designers. Furthermore, the sender-receiver problem which may arise during the communication process between customers and designers is a further reason leading to the misconception of customer affective needs (Blecker and Kreutler, 2004).

Kansei Engineering has been developed to deal with customers' subjective impressions (called Kansei in Japanese) regarding a product (Nagamachi, 1989). Using Kansei words, the customers are guided to express their affective needs, their feelings, and their emotional states. These emotional and sensory wants are then translated into perceptual design elements of the product (Nagamachi, 1996). While Kansei words excel in describing affective needs, the mapping relationships between Kansei words and design elements are often not clearly available in practice. Designers are often not aware of the underlying coupling and interrelationships among various design elements with regard to the achievement of customers' affective satisfaction. Clausing (1994) discerns customer needs and product specifications and points out that the mapping problem in between is the key issue in “design for customers”.

In addition, there rarely exists any definite structure of affective need information. Kansei words are usually expressed in abstract, fuzzy, or conceptual terms, leading to work on the basis of vague assumptions and implicit inference. A few researchers have enforced a hierarchical structure or an AND/OR tree structure for the articulation of customer needs, for example, the requirement taxonomy (Hauge and Stauffer, 1993), the customer attribute hierarchy (Yan et al., 2001), and the functional requirement topology (Tseng and Jiao, 1998). Nevertheless, the non-structured nature of affect information itself coincides with those difficulties in natural language processing (Shaw and Gaines, 1996).

Due to the above hindrances inherent in the Kansei mapping process, reusing knowledge from historical data suggests itself as a natural technique to facilitate the handling of affective need information, as well as tradeoffs among many design elements. To this end, this section proposes to apply data mining techniques to improve the identification of customers’ affective needs and the mapping of these needs to affective design elements. Based on association rule mining, this section develops an inference system for affective design decision support. The Kansei mining system utilizes valuable information latent in customers’ impressions on existing affective designs.

6.2.1. Problem Formulation

As shown in Figure 6-6, affective design involves a mapping process from affective needs in the customer domain to perceptual design elements in the design domain. It illustrates how a designer may achieve affective design and how the customer of the product will perceive and react. In general, customer affective needs can be described using a set of

Kansei words, $F \equiv \{f_m^* \mid m = 1, \cdots, M\}$, where $f^* =:: f_m^* \in F_m^*$. Suppose that there are multiple market segments, $S \equiv \{s_i \mid i = 1, \cdots, I\}$, each containing homogeneous customers. The customers in each segment comprise a set, $C_i \equiv \{c_1, c_2, \cdots, c_{N_i}\}$, where N_i denotes the total number of customers involved in i-th segment. For each segment, the affect information of a particular customer, $c_{n_i} \in C_i \mid \exists n_i \in [1, \cdots, N_i]$, can be depicted as a vector of certain Kansei words, for example, $\overline{f_{n_i}^*} \equiv [f_2^*, f_4^*, \cdots, f_8^*]$, where f_2^* refers to the 2-nd Kansei word employed by customer c_{n_i}, f_4^* the 4-th Kansei word, and f_8^* the 8-th Kansei word. The entire population of customers' affective needs constitute a set, $F^* \equiv \{\overline{f_1^*}, \overline{f_2^*}, \cdots, \overline{f_{N_i}^*}\}$.

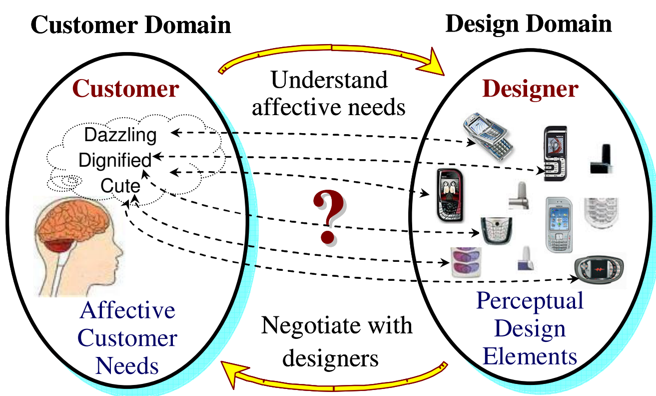


Figure 6-6 Mapping in affective design

Affective design yields many products that are desired by different customers. Each product is characterized by a set of perceptual design elements (DEs), $V \equiv \{v_q^* \mid q = 1, \cdots, Q\}$, where $v^* =:: v_q^* \in V_q^*$. All existing products comprise a set, $P \equiv \{p_1, p_2, \cdots, p_T\}$, where T refers to the total number of products. The specification of a particular product, $p_t \in P \mid \exists t \in [1, \cdots, T]$, can be represented as a vector of certain DEs, for example,

$\overline{v_t^*} \equiv \left[v_2^*, v_3^*, \cdots, v_6^*\right]$, where v_2^* means that product p_t involves the 2-nd DE, v_3^* the 3-rd DE, and v_6^* the 6-th DE. All the instances of DEs comprise a set, $V^* \equiv \left\{\overline{v_1^*}, \overline{v_2^*}, \cdots, \overline{v_T^*}\right\}$.

Differentiation between the customer domain (F^*) and the design domain (V^*) is consistent with the fact that customers' affective impressions are associated with products, rather than individual DEs. The customers do not know what their affective needs mean by mapping to specific DEs. The mapping relationship between customer affective needs and perceptual design elements is thus noted as $F^* \Rightarrow V^*$, where an association rule, $\Rightarrow$, indicates an inference from the precedent (F^*) to the consequence (V^*). All association rules constitute the knowledge base for the mappings from Kansei words to DEs, $\Lambda = \left\langle f_m^* \Rightarrow v_q^* \right\rangle$.

6.2.2. Kansei Mining

Figure 6-7 illustrates the architecture of the Kansei mining system, which consists of four modules, namely Kansei database construction, Kansei mining, goodness evaluation, and rule refinement and presentation. First, a relational database is established to document all target data extracted from past sales records and previous product specifications. All records are assorted by affective needs, design elements, and Kansei words. Then the Kansei mining procedure is initiated to search for interesting patterns. From Kansei mining, many useful rules are generated. Then goodness evaluation is enacted to justify the quality of rules with respect to individual segments. Finally, the rule refinement and presentation module comes into play to identify the most relevant and valuable rules and accordingly constructs the knowledge base.

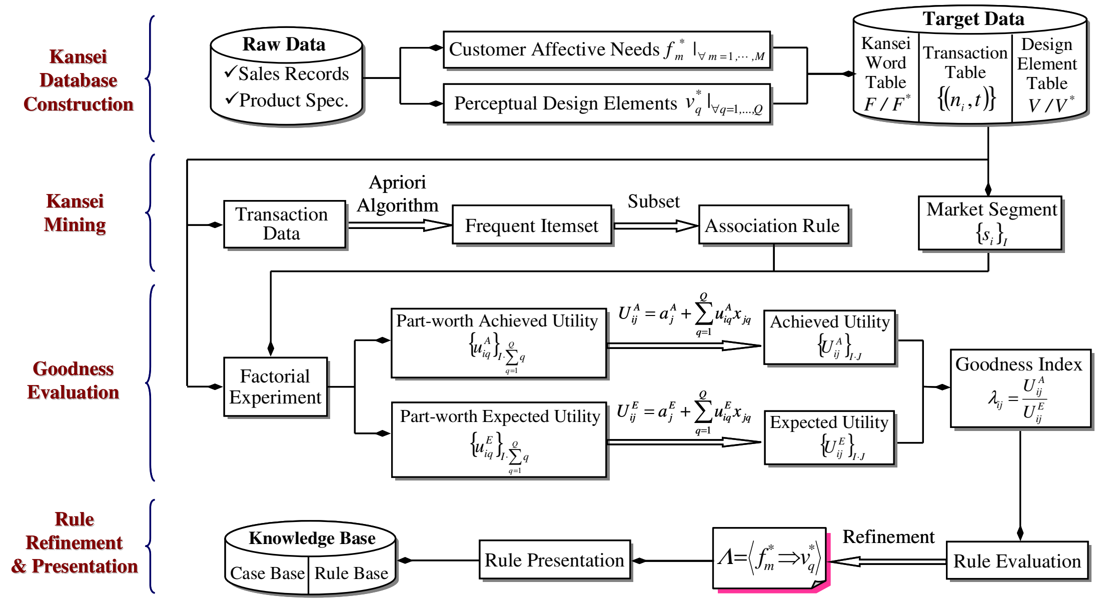


Figure 6-7 Kansei mining system architecture

6.2.2.1. Kansei database construction

Before proceeding to rule mining of data sets, raw data must be preprocessed in order to be useful for knowledge discovery. For association rule mining, it is most important to establish the transaction records first and then find the items involved in each transaction. One-to-one relationships should be established among various fields in the same transaction. A relational database model is considered for the Kansei database, as it allows files to be related by means of a common field which makes the model flexible. Figure 6-8 shows the entity relationships among transaction data.

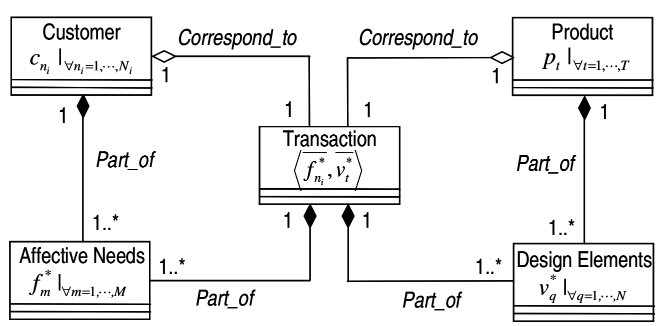


Figure 6-8 Organization of transaction data

The sales records are extracted from the company's legacy databases and are stored in the need information table labeled with the customer ID. The perceptual design elements are identified from previous product specifications and are stored in the product document table labeled with the product ID. The affective need information table thus contains all transaction records that entail the translation of customers' affect information to Kansei words. Kansei words are identified a priori from customer needs based on market research. Kansei words are mostly adjectives and sometimes nouns. A mobile phone customer, for example, may use such Kansei words as "comfortable", "highly qualified" and "cute" to articulate his/her subjective impression on a particular design that comprises a few perceptual design elements. The product document table contains information about existing design elements that constitute various product styles. These two tables are related through customer-product pairs that relate each customer ID to a product ID used to meet this customer, thus embodying mapping transaction data from previous designed products.

6.2.2.2. Kansei mining

The general form of an association rule in Kansei mining is given as the following,

$$\begin{array}{l} \alpha_1 \wedge \alpha_2 \cdots \wedge \alpha_x \cdots \wedge \alpha_X \Rightarrow \beta_1 \wedge \beta_2 \cdots \wedge \beta_y \cdots \wedge \beta_Y \\ [Support = s\%;\ Confidence = c\%] \end{array}, \quad (40)$$

where $\exists \alpha_x \in \{f_m^*\}_M \mid \forall x = 1, \cdots, X \leq M$, $\exists \beta_y \in \{v_q^*\}_Q \mid \forall y = 1, \cdots, Y \leq Q$, the meanings of the association rule in Eq. (40), as well as $s\%$ and $c\%$, are the same with those discussed in Chapter 4.

6.2.2.3. Goodness evaluation

Each association rule indicates a particular correspondence between certain Kansei words and a few design elements. Such a correspondence can be useful to suggest the underlying inference mechanism of affective design. Therefore, the goodness of each association rule has to be evaluated in order to find relevant and valuable mapping patterns. This is elaborated more in Section 6.2.3.

6.2.2.4. Rule refinement and presentation

Based on the evaluation results, the associated rules are refined to keep the most meaningful rules in the knowledge base in the form of either case bases or rule bases. The characteristics of each segment should also be explored based on the rules and the related support and confidence levels. Moreover, the causality of original association rules are defined for single DE options, as the precedent of each rule is a subset of $\{f_m\}$ and the consequence of each rule is a subset of $\{(n_i, t)\}$ per se. Nevertheless, inference relationships do exist in various combinations of more DE options. This means there is a need for generating combinatorial rules. To solve such a rule refinement problem, the Kansei mining system adopts an equivalence class method proposed by ChangChien and Lu (2001). Finally, users can retrieve all of the rules stored in the knowledge base to understand the mappings of affective needs to DEs clearly, to gain insights into the consequences of diverse customer preferences on different product images, and thus to justify the proper specification of product offerings in terms of perceptual features.

6.2.3. Goodness Evaluation for Association Rule Refinement

One of the challenges of association rule mining lies in the decision of thresholds, i.e., the minimum support and minimum confidence levels. Generally, allowing low levels of the thresholds may produce overwhelmed information; however, using too strict threshold levels may result in possible omission of useful mapping patterns. It is difficult to determine appropriate parameters for granularity, especially at the stage when the underlying patterns are still unknown. While arbitrary decisions for such parameters are deemed to be improper, most practitioners rely on the conjecture of domain experts.

In the case of Kansei mining, it is more preferable to adopt a two-step approach to generate the most promising rule patterns. At first, a set of raw rules are generated by specifying low values for the support and confidence thresholds. Low threshold levels warrant there are enough raw rules are yielded. The fact is that all rules generated using higher threshold levels are de facto subsets of the rule sets generated from less strict mining. And then, these raw rules are evaluated according to their goodness and are thus refined by discarding poor rules. Such a two-step approach circumvents the difficulties in justifying reasonable support and confidence thresholds, and thereby helps to identify meaningful rules.

6.2.3.1. Goodness index

It is necessary to choose the right criterion of goodness for rule refinement. Corresponding to certain customer needs represented as a bundle of Kansei words, the designer provides a bundle of design elements considered most approximate to meet the customer's expectation. From the designer's viewpoint, a customer's affective satisfaction can be interpreted as the customer's expected utility measured based on the customer's perceived benefits embodied in a combination of Kansei words. Nevertheless, from the customer's perspective, his/her perceived benefits may vary when designers deliver different bundles of design elements. This implies that the achieved utility of a design in terms of design elements is different from the original customer's expected utility in terms of Kansei words, although these design elements are supposed to be mapped from the specified Kansei words. The customer's perceived benefits from delivered design elements constitute the achieved utility, indicating what they really gain.

Therefore, the difference between the expected and achieved utilities reveals the degree of a customer's affective satisfaction. The theory of modern service marketing suggests that the more difference between the customers' gained service level and their expected service level, the more satisfied they are (Zeithaml and Bitner, 2001). More delight can even be created by achieving more than the expected utility (Kano et al., 1984). Such a difference further explains to what extent offering certain design elements can fit the customers' affective needs. As a result, a goodness index for mapping rules is introduced as the ratio of the achieved utility to the expected utility. The higher value of the ratio, the higher the quality of the rule will be. Such a ratio-based index is advantageous over the conventional approaches based on weighted sum. It enables a dimensionless measure of relative magnitude, in addition to overcoming the tedious issue of determining importance weights.

6.2.3.2. Segment-level goodness evaluation

Due to the heterogeneous nature of customer needs, measuring the customer perceived utility is difficult. For every two customers whose needs differ from each other, their appreciation of the benefits gained from the same product design may be distinct. In practice, companies always provide diverse products to accommodate different customers. For a product that is to serve certain customer needs, the perceived benefits may be less for those customers with dissimilar requirements; hence the average perceived benefit of a design is dominated by the majority of similar customer needs. This may distort the evaluation of a design if considering disparate customers at the same time. Market segmentation has convinced us that groups of customers with similar needs are likely to present a more homogeneous response to products and marketing programs (Kotler, 1994). As a result, rule refinement should be implemented at the segment level. That means both expected and achieved utilities should be measured according to the customers belonging to the same segment. Within the same segment, customer affective needs and rule patterns are similar.

Assume that there exist multiple market segments, $S \equiv \{s_i \mid i = 1, \cdots, I\}$. For each segment, s_i, a number of J raw rules are generated from the first step of Kansei mining. The customer's perceived benefits of DEs suggested by the j-th rule are measured as the achieved utility, $\{U_{ij}^A\}_{I \cdot J}$, corresponding to the customer's expected utility of those Kansei words in relation to the j-th rule, $\{U_{ij}^E\}_{I \cdot J}$. Suppose that there are L transaction records involved in the transaction database. Each transaction record comprises two item-sets, i.e., F^* and V^*. For each segment, s_i, the customer's expected utility for the item-set F^* involved in l-th transaction, where $l = 1, \cdots, L$, is represented by $\{U_{il}^F\}_{I \cdot L}$, and the customer's expected utility for the item-set V^* involved in l-th transaction is represented by $\{U_{il}^V\}_{I \cdot L}$.

A number of procedures for simultaneously performing market segmentation and calibrating segment-level part-worth utilities have been developed in recent years (Wedel and Kamakura, 1998). Among many methods, conjoint analysis has proven to be an effective means to estimate individual level part-worth utilities associated with individual product attributes (Green and Krieger, 1985). This research thus applies conjoint analysis to determine the expected and achieved utilities. A goodness index is computed as the following,

$$\lambda_{ij} = \frac{U_{ij}^A}{U_{ij}^E}, \tag{41a}$$

$$s.t.\quad U_{ij}^{A} = a_{j}^{A} + \sum_{q=1}^{Q} u_{iq}^{A} x_{jq}, \qquad \forall i \in \{1,\cdots,I\},\ \forall j \in \{1,\cdots,J\}, \tag{41b}$$

$$U_{ij}^{E} = a_{j}^{E} + \sum_{q=1}^{Q} u_{iq}^{E} x_{jq}, \qquad \forall i \in \{1,\cdots,I\},\ \forall j \in \{1,\cdots,J\}, \tag{41c}$$

$$U_{il}^{V} = U_{il}^{F} = a_{l} + \sum_{q=1}^{Q} u_{iq}^{E} y_{lq}, \qquad \forall i \in \{1,\cdots,I\},\ \forall l \in \{1,\cdots,L\}, \tag{41d}$$

$$x_{jq}, y_{lq} \in \{0,1\}, \qquad \forall j \in \{1,\cdots,J\},\ \forall l \in \{1,\cdots,L\},\ \forall q \in \{1,\cdots,Q\}, \tag{41e}$$

where λ_{ij} indicates the goodness of the j-th rule for segment s_i, U_{ij}^{A} denotes the achieved utility of the j-th rule with respect to segment s_i; U_{ij}^{E} stands for the expected utility of the j-th rule for segment s_i; U_{il}^{F} represents the total utility of all Kansei words involved in the l-th transaction for segment s_i; U_{il}^{V} is the total utility of all design elements included in the l-th transaction for segment s_i; u_{iq}^{A} means the achieved part-worth utility of the q-th design element for segment s_i; u_{iq}^{E} represents the expected part-worth utility of segment s_i in relation to the q-th design element; and constants a_{j}^{A}, a_{j}^{E} and a_{l} are respective intercepts.

Equation (41a) is to measure the goodness of the j-th rule, that is, to what extent the design elements involved in this rule fit the customer's expected utility. Equations (41b) and (41c) refer to the procedure of conjoint analysis – ensure that the composite utilities to be constructed from part-worth utilities of individual design elements, $\{v_{q}^{*}\}_{Q}$. Equation (41d) indicates that the customer expectations embodied in diverse customer needs are modeled as the expected part-worth utilities of individual design elements, $\{v_{q}^{*}\}_{Q}$. Constraint (41e) represents a binary restriction, where x_{jq} is a binary variable such that $x_{jq} = 1$ if the q-th design element is contained in j-th rule, and $x_{jq} = 0$ otherwise; and y_{lq} is a binary variable such that $y_{lq} = 1$ if the q-th design element is contained in l-th transaction, and $y_{lq} = 0$ otherwise.

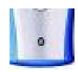
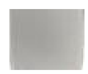
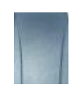
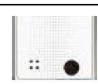
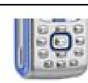
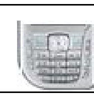
6.2.4. Case Study

The potential of Kansei mining has been tested in a company that produces a large variety of mobile phones. The company has conducted extensive market studies and competition analyses and projected the trends of design technologies in the business sector concerned. The historical data about the customer affective needs of mobile phones are assorted according to well-known Kansei words related to mobile phones (Khalid and Helander, 2004). As shown in Table 6-3, a total number of 15 Kansei words are used to describe affect information as perceived by different mobile phone users. Based on existing designs, a total of 23 perceptual design elements are extracted, as shown in Table 6-4.

Table 6-3 Kansei words for mobile phones

$f_m^* \mid \forall m = 1,\cdots,M$	Description	Code	$f_m^* \mid \forall m = 1,\cdots,M$	Description	Code
f_1^*	Portable	F1	f_9^*	Comfortable	F9
f_2^*	Sturdy	F2	f_{10}^*	Dazzling	F10
f_3^*	Enjoyable	F3	f_{11}^*	Mature	F11
f_4^*	Dignified	F4	f_{12}^*	Fashionable	F12
f_5^*	Cheerful	F5	f_{13}^*	Friendly	F13
f_6^*	Natural	F6	f_{14}^*	Cute	F14
f_7^*	Delightful	F7	f_{15}^*	Futuristic	F15
f_8^*	Stimulating	F8			

Table 6-4 Perceptual design elements for mobile phones

Code	V1	V2	V3	V4
v_i				
Code	V5	V6	V7	V8
v_i				
Code	V9	V10	V11	V12
v_i				
Code	V13	V14	V15	V16
v_i				
Code	V17	V18	V19	V20
v_i				
Code	V21	V22	V23	
v_i				

It is interesting to notice the difference between the customers' and designers' views on affective design of mobile phones. What customers really perceive is how they feel about the impression of a particular mobile phone design. Their affective needs are expressed in their own language (Kansei words). It is in the design domain where the affective aspect of a mobile phone is interpreted in terms of individual design elements. There is a practical need to fill the gap between the customers' expectations in the customer domain and product fulfillment in the design domain.

6.2.4.1. Transaction database

The set of Kansei words are stored in the affect information database, while perceptual design elements are stored in the product specification database. These two databases are interrelated with each other according to customers' choices of mobile phones. The target data are extracted from previous customer need information and product specifications and are organized into a transaction database, as shown in Table 6-5. Each transaction record indicates which design elements are used for fulfilling a customer's affective expectation.

Table 6-5 Transaction database

Record TID		Kansei Words $\overline{f_{n_i}^*} \equiv \left[f_m^* \mid \forall m = 1,\ldots,M\right]$	Design Elements $\overline{v_t^*} \mid \forall t = 1,\cdots,T$
Segment 1	T001	F1, F2, F6, F11, F13	V1, V5, V8, V10, V12, V15, V18, V20
	T002	F1, F3, F6, F7, F11, F13	V3, V5, V12, V15, V18, V20, V21
	T004	F2, F3, F6, F7, F9, F11, F13, F14	V1, V3, V5, V10, V13, V21, V22
	…	…	…
	T024	F6, F7, F9, F13, F14	V3, V5, V8, V9, V18, V20, V22
	T025	F3, F7, F11, F13, F14	V3, V6, V10, V13, V18, V20, V21, V22
	T029	F1, F6, F7, F9, F13, F14	V3, V5, V8, V13, V15, V18, V20, V22
Segment 2	T003	F5, F8, F12, F15	V2, V4, V6, V7, V9, V17, V19
	T005	F3, F4, F8, F10, F12, F15	V2, V6, V7, V9, V16, V17, V19, V21
	T006	F5, F8, F12, F15	V2, V4, V6, V7, V9, V17, V19
	…	…	…
	T027	F3, F4, F5, F8, F10, F12	V2, V4, V6, V7, V10, V11, V17, V19
	T028	F3, F5, F8, F12, F15	V2, V4, V6, V7, V10, V17, V21
	T030	F4, F5, F10, F12	V6, V10, V11, V13, V17, V19

For illustrative simplicity, only 30 out of hundreds of transaction records are used in the case study here. As shown in Table 6-5, the set of Kansei words for each customer indicates the customer's affective needs for his/her choice of mobile phones, which are described as a particular instance of the subset of $F = \{f_m^*\}_M$. Among the 30 mobile phone designs provided to satisfy the 30 customers, the design elements used in each design are represented as specific instances of the DE vector, $\left[v_q^*\right]_Q$.

6.2.4.2. Association rule mining

As shown in Table 6-5, the 30 transaction records are organized in two segments, s_1 and s_2, which are identified based on established market research of the company. Segment s_1 includes customer records 1, 2, 4, 7, 8, 11, 12, 14, 15, 18, 21, 24, 25 and 29. Segment s_2

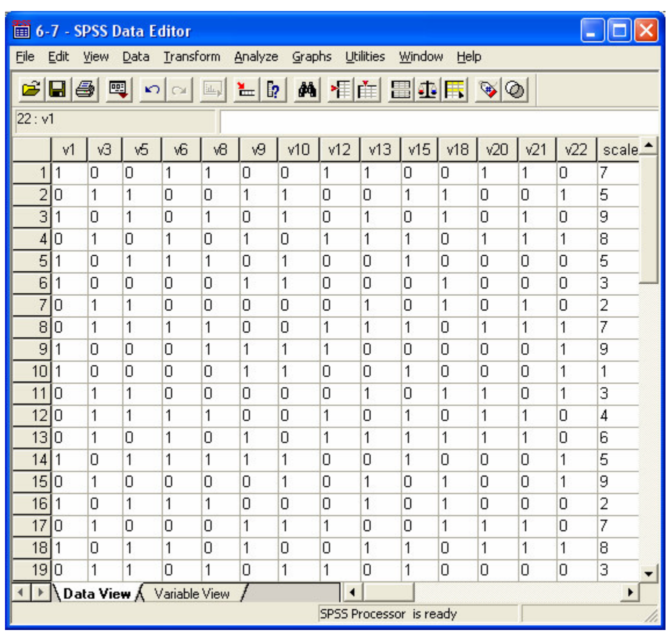
consists of customers 3, 5, 6, 9, 10, 13, 16, 17, 19, 20, 22, 23, 26, 27, 28 and 30. A data mining tool, Magnum Opus (Version 2.0, www.rulequest.com), is employed to find the mapping relationships between the Kansei word item-set and the design element item-set for each segment. The mining process runs two times and terminates with two sets of rules containing 265 and 173 association rules for segments 1 and 2, respectively. For illustrative simplicity, only 20 rules are presented here for each segment, as shown in Table 6-6.

Table 6-6 Association rules produced by Kansei mining

Rule		Inference Relationship	Support	Confidence
Segment 1	1	portable ⇒ V15	0.633	0.323
	2	portable ⇒ V12	0.633	0.267
	3	delightful ⇒ V3	0.300	0.289
	4	enjoyable ⇒ V21	0.329	1.000
	5	mature ⇒ V10	0.233	0.322
	6	mature ⇒ V18	0.233	0.368
	7	natural ⇒ V18	0.267	1.000
	8	delightful ⇒ V22	0.300	0.323
	9	comfortable ⇒ V22	0.267	0.933
	10	portable ⇒ V5 & V15 & V12	0.633	0.315
	11	delightful ⇒ V13 & V3	0.300	0.267
	12	cute ⇒ V12	0.600	1.000
	13	delightful & cute ⇒ V3 & V22	0.264	0.875
	14	natural & mature & friendly ⇒ V18 & V20 & V10	0.206	0.764
	15	delightful & comfortable & cute ⇒ V22 & V3 & V12	0.263	0.872
	16	natural & delightful & friendly ⇒ V18 & V20 & V3 & V8	0.212	0.864
	17	mature & natural & friendly & comfortable ⇒ V18 & V20 & V22 & V10	0.200	0.664
	18	cute & portable ⇒ V12 & V5 & V15	0.526	0.835
	19	mature & enjoyable & sturdy ⇒ V18 & V20 & V8 & V10	0.200	0.763
	20	natural & portable & friendly ⇒ V12 & V10 & V5 & V15	0.200	0.625
Segment 2	1	enjoyable ⇒ V21	0.600	1.000
	2	fashionable ⇒ V2	0.467	0.433
	3	fashionable ⇒ V7	0.467	0.226
	4	dignified ⇒ V17 & V19	0.700	0.227
	5	dignified ⇒ V19	0.700	0.323
	6	fashionable ⇒ V6	0.467	0.200
	7	cheerful ⇒ V7 & V4	0.627	0.375
	8	dignified ⇒ V17	0.700	0.289
	9	stimulating ⇒ V6	0.362	0.362
	10	cheerful ⇒ V6 & V2	0.627	0.482
	11	dazzling ⇒ V17 & V9	0.533	0.875
	12	stimulating ⇒ V7	0.362	0.325
	13	cheerful & stimulating ⇒ V7 & V4	0.300	1.000
	14	dazzling & cheerful ⇒ V17 & V6 & V2	0.206	0.764
	15	enjoyable & dignified ⇒ V17 & V19 & V21	0.233	0.825
	16	dignified & dazzling ⇒ V17 & V19 & V9	0.267	0.923
	17	dazzling & fashionable & stimulating ⇒ V17 & V2 & V9	0.327	0.671
	18	futuristic & dignified & enjoyable & stimulating ⇒ V17 & V19 & V21	0.300	0.648
	19	enjoyable & futuristic & cheerful ⇒ V19 & V21 & V2	0.253	0.876
	20	dazzling & futuristic ⇒ V17 & V19	0.325	0.712

6.2.4.3. Goodness evaluation

The achieved utility of an association rule is derived from the customer's perceived utility in terms of related design elements to this rule. Customer perceived utilities are determined based on conjoint analysis. Conjoint analysis starts with the construction of testing choice sets. Orthogonal experiments are designed using the Orthogonal Array Selector provided by SPSS software (www.spss.com). A total of 36 and 27 orthogonal testing choice sets are generated for segments s_1 and s_2, shown in Figures 6-9 and 6-10, respectively. With these choice sets, two fractional factorial experiments are designed to explore the achieved utility of every design element for each segment. The results of respective experiment designs are explained in Tables 6-7 and 6-8. For instance, a value of 1 in columns 2-15 of Table 6-7 indicates that v_q^* is involved in the choice sets, and 0 means that it is not selected. The last column of Table 6-7 collects the perceived benefits by the respondents.

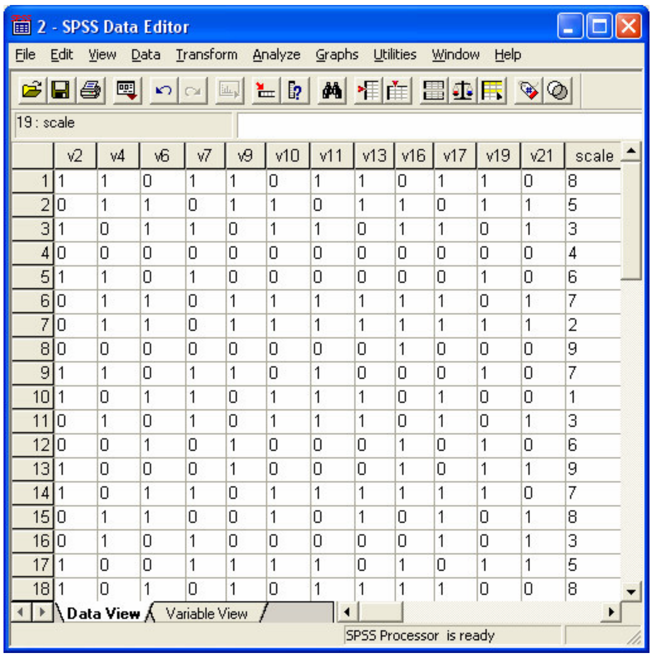


	v1	v3	v5	v6	v8	v9	v10	v12	v13	v15	v18	v20	v21	v22	scale
1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	7
2	0	1	1	0	0	1	1	0	0	1	1	0	0	1	5
3	1	0	1	0	1	0	1	0	1	0	1	0	1	0	9
4	0	1	0	1	0	1	0	1	1	1	0	1	1	1	8
5	1	0	1	1	1	0	1	0	0	1	0	0	0	0	5
6	1	0	0	0	0	1	1	0	0	0	1	0	0	0	3
7	0	1	1	0	0	0	0	0	1	0	1	0	1	0	2
8	0	1	1	1	1	0	0	1	1	1	0	1	1	1	7
9	1	0	0	0	1	1	1	1	0	0	0	0	0	1	9
10	1	0	0	0	0	1	1	0	0	1	0	0	0	1	1
11	0	1	1	0	0	0	0	0	1	0	1	1	0	1	3
12	0	1	1	1	1	0	0	1	0	1	0	1	1	0	4
13	0	1	0	1	0	1	0	1	1	1	1	1	1	0	6
14	1	0	1	1	1	1	1	0	0	1	0	0	0	1	5
15	0	1	0	0	0	0	1	0	1	0	1	0	0	1	9
16	1	0	1	1	1	0	0	0	1	0	1	0	0	0	2
17	0	1	0	0	0	1	1	1	0	0	1	1	1	0	7
18	1	0	1	1	0	1	0	0	1	1	0	1	1	1	8
19	0	1	1	0	1	0	1	1	0	1	0	0	0	0	3

Figure 6-9 Testing choice sets for segment 1

Table 6-7 Response surface experiment design for segment 1

	Conjoint Test (Segment 1)														Preference Scale
Choice	V1	V3	V5	V6	V8	V9	V10	12	V13	V15	V18	V20	V21	V22	1 … 9
1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	7
2	0	1	1	0	0	1	1	0	0	1	1	0	0	1	5
3	1	0	1	0	1	0	1	0	1	0	1	0	1	0	9
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
34	0	0	0	1	0	0	0	1	0	0	0	1	0	0	4
35	1	1	1	0	1	1	1	0	1	1	1	0	1	1	5
36	0	1	1	1	0	1	1	1	0	1	1	1	0	1	2



	v2	v4	v6	v7	v9	v10	v11	v13	v16	v17	v19	v21	scale
1	1	1	0	1	1	0	1	1	0	1	1	0	8
2	0	1	1	0	1	1	0	1	1	0	1	1	5
3	1	0	1	1	0	1	1	0	1	1	0	1	3
4	0	0	0	0	0	0	0	0	0	0	0	0	4
5	1	1	0	1	0	0	0	0	0	0	1	0	6
6	0	1	1	0	1	1	1	1	1	1	0	1	7
7	0	1	1	0	1	1	1	1	1	1	1	1	2
8	0	0	0	0	0	0	0	0	1	0	0	0	9
9	1	1	0	1	1	0	1	0	0	0	1	0	7
10	1	0	1	1	0	1	1	1	0	1	0	0	1
11	0	1	0	1	0	1	1	1	0	1	0	1	3
12	0	0	1	0	1	0	0	0	1	0	1	0	6
13	1	0	0	0	1	0	0	0	1	0	1	1	9
14	1	0	1	1	0	1	1	1	1	1	1	0	7
15	0	1	1	0	0	1	0	1	0	1	0	1	8
16	0	1	0	1	0	0	0	0	0	1	0	1	3
17	1	0	0	1	1	1	1	0	1	0	1	1	5
18	1	0	1	0	1	0	1	1	1	1	0	0	8

Figure 6-10 Testing choice sets for segment 2

Table 6-8 Response surface experiment design for segment 2

	Conjoint Test (Segment 2)												Preference Scale
Choice	V2	V4	V6	V7	V9	V10	V11	V13	V16	V17	V19	V21	1 … 9
1	1	1	0	1	1	0	1	1	0	1	1	0	8
2	0	1	1	0	1	1	0	1	1	0	1	1	5
3	1	0	1	1	0	1	1	0	1	1	0	1	3
…	…	…	…	…	…	…	…	…	…	…	…	…	…
25	0	0	1	0	0	1	0	0	1	0	0	1	5
26	1	0	0	1	0	0	1	0	0	1	0	0	2
27	0	1	0	0	1	0	0	1	0	0	1	0	6

For the two segments, a total of 14 and 16 customers are selected to act as the respondents, respectively. Each respondent is asked to evaluate 36 (or 27) choices one by one based on a 9-point scale, where "9" means the customer perceives the most benefit and "1" the least. This results in $14 \times 36 = 504$ and $16 \times 27 = 432$ groups of data for two segments, respectively. For each respondent in segment s_1 or s_2, a total of 504 or 432 regression equations are obtained by interpreting his/her original choice data as a binary instance of the part-worth utility. Each regression corresponds to a bundle of design elements and indicates the achieved benefit perceived by the respondent. By running multivariate regression, where DE is encoded as 1 if it is contained in the regression model, 0 otherwise, the achieved part-worth benefits for the respondent are derived. Averaging the achieved part-worth benefits of all respondents within one segment, a segment-level achieved utility is derived for each individual design element.

Likewise the expected part-worth utility of each design element is derived based on the conjoint analysis procedure. Rather than relying on choice set construction, the respondents are asked to evaluate their perceived benefit of each Kansei word contained in a transaction record. The design elements involved in this transaction suppose to deliver a utility as much as what the respondent expects using Kansei words. Thus, all the transaction records become the choice sets, where v_q^* is encoded as 1 if it is contained in the transaction, and 0 otherwise. The customers' expected benefits are used as the assessment criteria for each choice in the fractional factorial experiment. Table 6-9 shows the results of the respective expected and achieved part-worth utilities of every design element within two segments.

Table 6-9 Part-worth utilities for individual design elements

Segment 1			Segment 2		
DE	Expected Utility	Achieved Utility	DE	Expected Utility	Achieved Utility
V1	0.08	0.05	V2	1.27	1.65
V3	1.67	1.95	V4	1.37	0.73
V5	1.86	1.13	V6	1.41	0.82
V6	0.03	0.04	V7	1.14	0.61
V8	1.28	1.31	V9	0.83	1.24
V9	0.11	0.14	V10	0.13	0.11
V10	0.86	0.91	V11	0.12	0.07
V12	1.73	1.04	V13	0.04	0.06
V13	1.46	1.45	V16	0.05	0.07
V15	0.93	0.31	V17	1.23	1.56
V18	0.93	0.96	V19	1.24	1.85
V20	1.28	1.25	V21	0.82	0.87
V21	1.47	1.45			
V22	1.12	1.26			

Based on the part-worth utilities, the achieved utility, U_{ij}^{A}, and the expected utility, U_{ij}^{E}, of every association rule for each segment are composed according to Eqs. (41b-e) and shown in Table 6-10. Accordingly, the corresponding goodness index for each rule is calculated using Eq. (41a). The results are shown in Figure 6-11.

6.2.4.4. Rule refinement

As shown in Figure 6-11, high goodness measures indicate a good mapping relationship in terms of the achievement of the customer's affective satisfaction, whereas a low value indicating poor rules. Among those good mapping rules, some designs (for example Rule 3 for segment 1 and Rule 5 for segment 2) outperform the customers' original expectations ($\lambda_{13} = 1.17 > 1$ and $\lambda_{25} = 1.49 > 1$), which is consistent with the wisdom suggest by Kano diagram (Kano et al., 1984). Such designs are considered "delighters" for customer satisfaction, in addition to those "must-have" designs whose achievements fall into $\lambda^{*} \leq \lambda_{ij} \leq 1$. On the other hand, those rule patterns yielding poor goodness assumed by Table

6-10 (for example, $\lambda_{11} = 0.33$) do not contribute much to customers' satisfaction, and thus should be discarded. A threshold of $\lambda^* = 0.7$ is determined a priori by domain experts. As a result, Rules 1, 2, 10, 12, 18, and 20 are ignored for segment 1, and Rules 3, 6, 7, 9, 12, and 13 are discarded for segment 2. The results of rule refinement are shown in Table 6-11.

Table 6-10 Result of goodness evaluation

Segment 1				Segment 2			
Rule #	v_q^*	U_{1j}^A	U_{1j}^E	Rule #	v_q^*	U_{2j}^A	U_{2j}^E
1	V15	0.31	0.93	1	V21	0.87	0.82
2	V12	1.04	1.73	2	V2	1.65	1.27
3	V3	1.95	1.67	3	V7	0.61	1.14
4	V21	1.45	1.47	4	V17, V19	3.41	2.47
5	V10	0.91	0.86	5	V19	1.85	1.24
6	V18	0.96	0.93	6	V6	0.82	1.41
7	V18	0.96	0.93	7	V4, V7	1.34	2.51
8	V22	1.26	1.12	8	V17	1.56	1.23
9	V22	1.26	1.12	9	V6	0.82	1.41
10	V5, V12, V15	2.48	4.52	10	V2, V6	2.47	2.68
11	V3, V13	3.4	3.13	11	V9, V17	2.8	2.06
12	V12	1.04	1.73	12	V7	0.61	1.14
13	V3, V12	2.99	3.4	13	V4, V7	1.34	2.51
14	V10, V18, V20	3.12	3.07	14	V2, V6, V17	4.03	3.91
15	V3, V12, V22	4.25	4.52	15	V17, V19, V21	4.28	3.29
16	V3, V8, V18, V20	5.47	5.16	16	V9, V17, V19	4.76	3.3
17	V10, V18, V20, V22	4.38	4.19	17	V2, V9, V17	4.45	3.33
18	V5, V12, V15	2.48	4.52	18	V17, V19, V21	4.28	3.29
19	V5, V10, V18, V20	4.25	4.93	19	V2, V19, V21	4.37	3.33
20	V5, V10, V12, V15	3.39	5.38	20	V17, V19	3.41	2.47

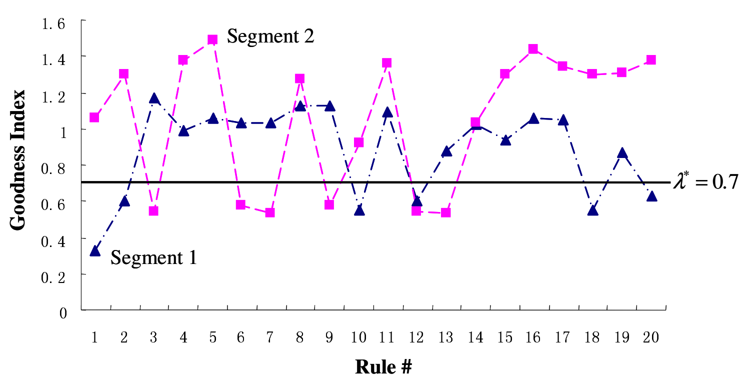


Figure 6-11 Comparison of goodness evaluation for two segments

Table 6-11 Refined rule sets for two segments

	Association Rule #
Segment 1	3, 4, 5, 6, 7, 8, 9, 11, 13, 14, 15, 16, 17, 19
Segment 2	1, 2, 4, 5, 8, 10, 11, 14, 15, 16, 17, 18, 19, 20

6.2.5. Validation for Affective Design Support

To validate the rationale of identified Kansei mapping relationships in support of affective design, a separate set of past designs are used for testing. Five transaction records from each segment are selected. Another 20 respondents for each segment are invited as the customers to evaluate these testing products (referred to as existing designs). Based on the original affective needs documented in respective transaction data, the Kansei mining system suggests another set of designs (referred to as inferred designs). Following the conjoint analysis procedure, these 40 respondents indicate their perceived utilities through Kansei word for the existing designs as well as the inferred designs. Then the expected utility of affective needs, the achieved utility of existing design, and the achieved utility of inferred design are derived for every original product in each segment. The support for affective design manifests itself through improvements in the achieved utility and goodness measure at both the product and segment levels.

Table 6-12 shows the part-worth expected and achieved utilities of every design element. These part-worth utilities are derived from the responses of 40 customers. As different groups of respondents are engaged, their perceived part-worth utilities may bear slight variation (e.g., Table 6-9 vs. Table 6-12). Table 6-13 summarizes the performances of the existing and inferred designs as perceived by the testing group of respondents. All inferred designs outperform the originally designed products. The maximal improvement of customer's perceived utilities reaches 27.59%, with a minimum of 0.916%. In terms of goodness measure, the improvement is as much as 27.48% maximum and 0.882% minimum. At the segment level, the overall performance is also improved. The improvements of the achieved utility and goodness measure for segment s_1 are 12.35% and 13.21%, respectively. The cohort performance of segment s_2 is also improved, with 10.89% and 10.65% for the utility and goodness measure, respectively. The reason for such improvement appears to be straightforward. All inferred designs are derived based on previous best practices encoded into association rules, whereas the original designs resulted from the rules-of-thumb by individual designers.

Table 6-12 Part-worth utilities perceived by testing groups

Segment 1			Segment 2		
DE	Expected Utility	Achieved Utility	DE	Expected Utility	Achieved Utility
V1	0.06	0.05	V2	3.72	1.63
V3	1.63	1.91	V4	1.42	0.81
V5	1.73	1.16	V6	1.43	0.75
V6	0.08	0.03	V7	3.17	1.06
V8	1.23	1.34	V9	0.85	1.28
V9	0.13	0.07	V10	0.07	0.08
V10	0.92	1.41	V11	0.13	0.11
V12	1.12	1.08	V13	0.06	0.04
V13	1.45	0.39	V16	0.11	0.09
V15	0.97	0.38	V17	1.17	1.52
V18	0.97	1.05	V19	1.31	1.92
V20	1.34	1.31	V21	0.58	1.21
V21	1.48	1.42			
V22	1.08	1.35			

Table 6-13 Performance comparison of design achievement

Transaction	TID	Affective Needs: Kansei Words	Affective Needs: Expected Utility	Existing Product: Design Elements	Existing Product: Achieved Utility	Existing Product: Goodness Index	Inferred Product: Design Elements	Inferred Product: Achieved Utility	Inferred Product: Goodness Index	Improvement (%): Utility	Improvement (%): Goodness Index
Segment 1	1	F2, F3, F7, F9, F14	7.67	V3, V5, V10, V15, V20, V22	7.52	0.98	V3, V8, V10, V15, V20, V22	7.70	1.004	2.393	2.449
	2	F6, F7, F9, F11, F13	5.76	V10, V13, V18, V20, V22	5.51	0.957	V3, V10, V18, V20, V22	7.03	1.220	27.59	27.48
	3	F6, F7, F11, F14	8.11	V8, V10, V12, V13, V15, V20, V22	7.26	0.895	V3, V10, V12, V15, V18, V20, V22	8.49	1.045	16.94	16.76
	4	F3, F6, F7, F11, F14	6.46	V3, V10, V12, V13, V20	6.1	0.944	V3, V10, V13, V18, V21	7.26	1.124	19.02	19.07
	5	F2, F3, F11, F13	7.67	V3, V5, V10, V18, V20, V22	8.19	1.068	V3, V8, V10, V18, V20, V22	8.37	1.091	2.198	2.154
				Segment Average	6.916	0.969		7.77	1.097	12.35	13.21
Segment 2	6	F3, F4, F10	6.74	V2, V6, V9, V19, V21	7.64	1.134	V2, V9, V17, V19, V21	7.71	1.144	0.916	0.882
	7	F4, F5, F10	5.33	V4, V6, V17, V19	5.00	0.938	V2, V6, V17, V19	5.82	1.092	16.4	16.42
	8	F8, F10, F12, F15	7.07	V4, V7, V9, V17, V19	6.59	0.932	V2, V7, V9, V17, V19	7.41	1.048	12.44	12.45
	9	F4, F5, F8, F10, F12	9.40	V2, V4, V6, V9, V17, V19	7.91	0.841	V2, V4, V7, V9, V17, V19	9.43	1.003	19.22	19.26
	10	F3, F4, F8, F10, F15	8.50	V4, V7, V9, V17, V19, V21	8.02	0.944	V2, V7, V9, V17, V19, V21	8.62	1.014	7.481	7.415
				Segment Average	7.032	0.958		7.798	1.06	10.89	10.65

6.3. Product Family Configuration Design

Developing product families has been well recognized as an effective means to achieve the economy of scale in order to accommodate increasing product variety across diverse market niches (Meyer and Utterback, 1993; Sundgren, 1999). In addition to leveraging the cost of delivering variety by reusing proven elements in a firm's activities and offerings, product family design (PFD) can offer a multitude of benefits including reduction in development risks and system complexity, improved ability to upgrade products, and enhanced flexibility and responsiveness of manufacturing processes (Sawhney, 1998).

PFD is often modeled as a type of configuration design, namely PFCD, which aims at selecting and arranging combinations of a set of predefined components/modules to generate an optimal mix of design alternatives subject to customer requirements and engineering or physical constraints. As the number of components/modules increases, the number of possible configuration design alternatives may be huge, and thus complete enumeration to obtain optimal design alternatives becomes numerically prohibitive (Tarasewich and Nair, 2001). Comparing with traditional calculus-based or approximation optimization techniques, genetic algorithms (GAs) have been proven to excel in solving combinatorial optimization problems (Steiner and Hruschka, 2002). However, either a specific GA or universally applicable GA has difficulties in dealing with the PFCD problem. This may stem from the complications inherent in the PFCD problem as elaborated next.

(1) Complexity of product family data. Instead of a collection of individual product variants, the organization of product family data needs to explicate the relationships between variants, i.e., deal with the product family rather than individual variants. Moreover, PFCD is implemented from both a commercial viewpoint and a technical viewpoint. Product variants thereby should be represented in terms of customer requirements, end-products, subassemblies, components, features and feature levels, as well as their relationships for engineering purposes. In the meantime, product variants propagate along the product structure by exploring the bill-of-materials (BOM). The vast and complex variants institute multiple levels of configuration and a large number of choices, and thus diverse individual configuration spaces need to be explored. Traditional GAs have difficulties in distinguishing configuration spaces and are not reusable in various configuration cases. This means that when configuration spaces change their contents according to diverse customer requirements, both the objective models and chromosome representation schemes need to be modified to

adapt to the varied problem. As such, traditional GAs are only suitable for individual PFCD scenarios, but not the entire PFCD space.

(2) Constraint handling. There are mainly two types of constraints involved in PFCD: compatibility constraints and selection constraints (Du et al., 2001). Compatibility constraints refer to the restrictions on choices of variants' contents (e.g., components, features, feature levels) and are generally described as IF-THEN rules. Selection constraints refer to customer requirements for variants' conditions (e.g., budget). Although a universally applicable GA based on universal encoding may be adapted to diverse PFCD scenarios, real problems are too complex to allow direct encoding, where the chromosome represents the original solution of a given problem as a whole (Gen and Cheng, 1997). For such complex problems as PFCD, a universally applicable GA often yields infeasible offspring due to the ineffectiveness in constraint handling (Kamrani and Gonzalez, 2003).

Inspired by the generic variety structure (GVS) (Jiao and Tseng, 1999), and the heuristic genetic algorithm discussed in Chapter 5, this section presents a generic genetic algorithm (GGA) for the PFCD problem. Distinguishing "generic" from "universal", the GGA does not attempt to encompass the entire solution within a single chromosome. Instead, the GGA is developed by formulating a generic encoding scheme, which adapts to diverse PFCD scenarios in accordance with a generic variety structure. An efficient constraint-handling strategy is incorporated into the GGA process to facilitate the generation of feasible offspring efficiently. The GGA enables the reusability of GAs along with the variation of configuration spaces in various PFCD scenarios thus improving the efficiency of PFCD problem-solving.

6.3.1. Configuration Space Formulation

Product family configuration design starts with creating a PFA (Product Family Architecture) generic variety structure (GVS) representation. Subsequently, customers are allowed to propose requirements according to customizable features represented in the GVS. A configuration space is then developed from the synthesized information embodied by a set of predefined modules and the customer requirements. All possible configuration alternatives are included in the configuration space. An optimization algorithm is then employed to produce the optimal configuration alternatives according to the objective function. As shown in Figure 6-12, within the GVS, all product variants of a family share a common structure. A combined decomposition/classification tree is adopted to represent functional classification from an abstract level to individual instances. In Figure 6-12, a node denotes a *P* variable while a leaf represents an instance of a *P* variable. The functional specification of a product family can be represented by a *P* vector, i.e., $P_{Family} = \{p_1, p_2, p_3, p_4, p_5\}$, and the specific specification of a product variant within this family is an instance of this *P* vector, e.g., $P_{Variant} = \{p^*_{111}, p^*_{121}, p^*_{212}, p^*_{312}, p^*_{411}\}$. The configuration constraints manifest themselves through restrictions on the combinations of the *P* vector instances and are expressed as a XOR (i.e., exclusive OR) relationship. For example, Figure 6-12 shows a size-compatible constraint among instances p^*_{122}, p^*_{321}, and p^*_{422}.

Consistent with the GVS, a configuration space is established as a hierarchical structure where a number of feasible configuration design alternatives, modules, candidates, features, design parameters, and their relationships are described within a single formalism. As shown in Figure 6-13, a configuration space is represented as an AND/OR graph. The configuration space is composed of N configuration design alternatives, each of which is configured by M modules. Each module contains a number of available candidates, among which only one can be chosen for final solutions. Each candidate is assumed to contain two functional features and two corresponding design parameters. The hierarchical structure makes it easy to identify multi-level configurations of subassemblies, intermediate parts, and component parts, as well as to explicate their interrelationships. Product variants can be identified along the spectrum of the GVS-based configuration model. Comparing with traditional approaches based on "enumeration" or "selection", which may work in a limited choice case, the GVS-based configuration model makes it possible to handle a large variety of variants involved in PFCD, and provides a concise way of "combination" for improving the efficiency of optimization.

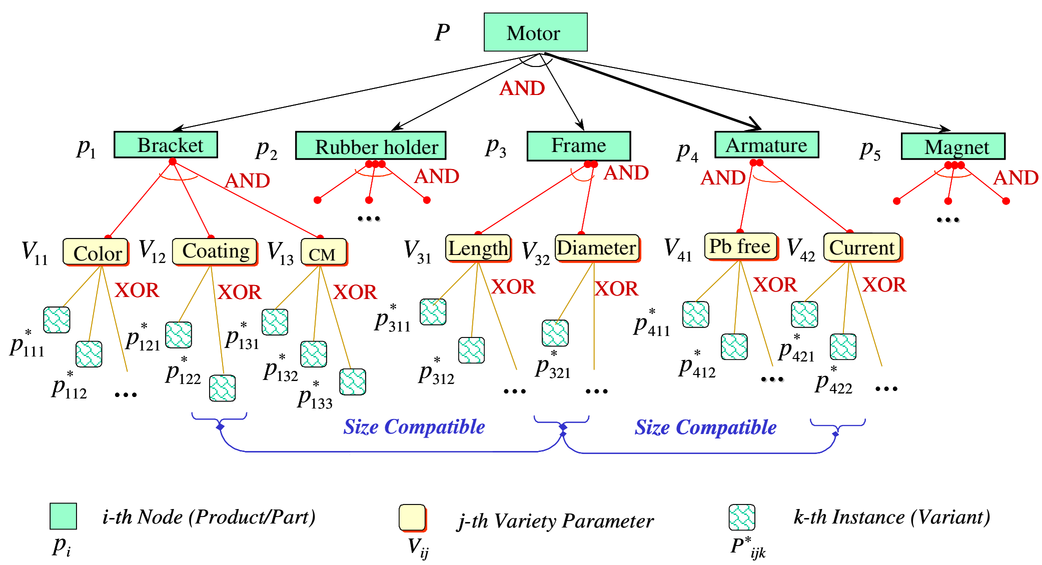


Figure 6-12 A generic structure for representing variety

6.3.2. Problem Formulation

Suppose a set of modules are identified, $M \equiv \{m_1, m_2, \cdots, m_K\}$. Each module, $m_k \mid \forall k \in [1, \cdots, K]$, may take on one out of a finite set of candidates,

$M_k^* \equiv \{m_{k1}^*, m_{k2}^*, \cdots, m_{kL_k}^*\}$. That is, $m_k =:: m_{kl}^* \mid \exists m_{kl}^* \in M_k^*$, where $l = 1, \cdots, L_k$, denotes the l-th candidate of m_k. Each module, $m_k \mid \forall k \in [1, \cdots, K]$, comprises a set of features, $F_k \equiv \{f_{kt} \mid t = 1, \cdots, T_k\}$. Each feature, $\forall f_{kt} \in F_k$, possesses a few levels, which are either discrete or continuous, i.e., $F_k^* \equiv \{f_{ktq}^* \mid q = 1, \cdots, Q_t\}$. A set of feasible configuration design alternatives, $A \equiv \{a_1, a_2, \cdots, a_I\}$, where $a_i = [m_{kl}^*]_K \mid \forall i \in [1, \cdots, I]; l \in [1, \cdots, L_k]$, are generated by choosing one of the candidates for certain modules, subject to satisfying specific configuration constraints.

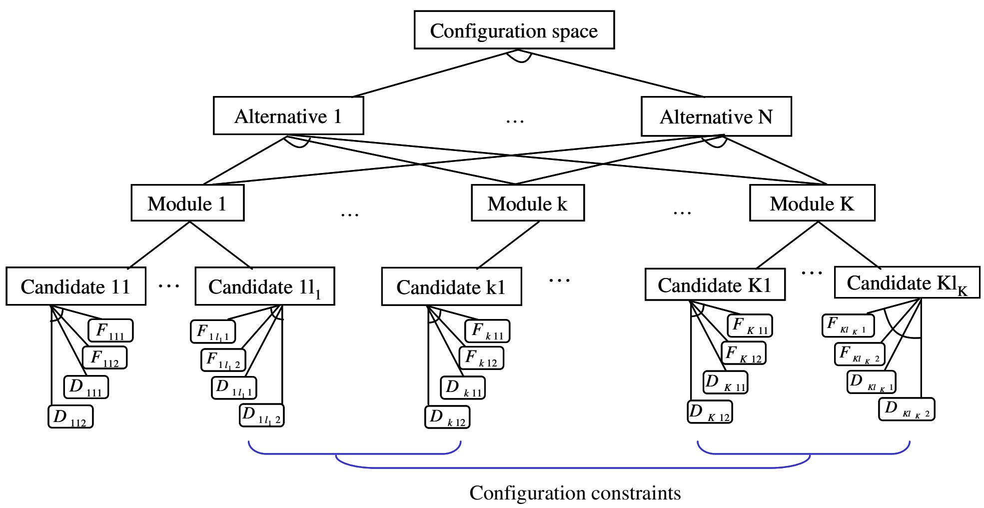


Figure 6-13 A configuration space

6.3.3. Optimization Model

Similar to the framework of product portfolio optimization, a maximizing shared surplus model is adopted for PFCD performance evaluation (as discussed in Chapter 5). Such a metric is advantageous over conventional metrics for PFCD, which focus more on engineering concerns. The objective function and involved issues are like those discussed in Chapter 5. Comparing with the product portfolio planning problem, where products are constructed directly from individual attributes, PFCD involves multiple levels of configurations including end-products, subassemblies, components, features, and feature levels.

6.3.4. Generic GA Design

The GAs are widely recognized owing its capability to produce acceptable solutions for combinatorial optimization problems involving a wide variety of configurations. Traditionally, a problem-specific encoding scheme is used to deal with a particular configuration space, where a unique optimization model is formulated. As a result, a specific GA can only be applied to solve a single PFCD scenario. The complexity of using a problem-specific GA is illustrated in Figure 6-14. Figure 6-14 shows three distinct PFCD cases where each GA-based PFCD case is a separate process.

The generic genetic algorithm (GGA) is developed to enable diverse configuration spaces by making use of a generic encoding scheme that originates from the GVS. The use of the GGA for PFCD is far more straightforward compared to the traditional GA. The complexity of using the GGA is denoted in Figure 6-15 where the three GGA-based PFCD cases follow a common process.

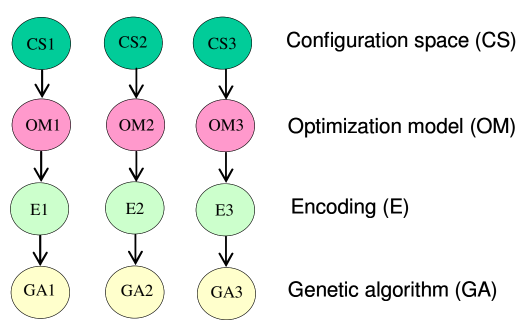


Figure 6-14 Problem-specific GAs for PFCD

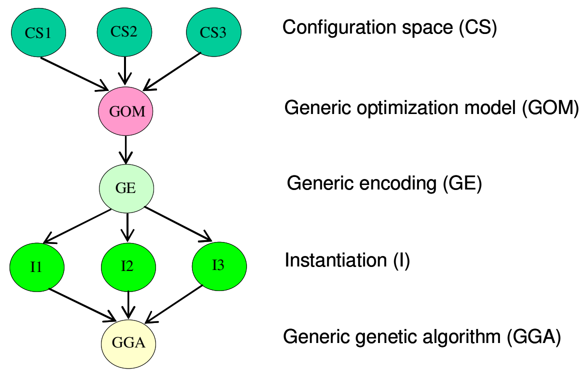


Figure 6-15 The GGA for PFCD

6.3.4.1. Generic encoding

In response to specific customer requirements, diverse configuration spaces need to be formulated. A generic encoding is able to characterize the variation of configuration spaces and diverse product variants. The generic encoding represents the PFCD problem using a finite-length string called a chromosome. Each fragment of the chromosome (i.e., substring) represents a module candidate contained in the product family. Each element of the string, called a gene, indicates a feature contained in the module candidate. The value assumed by a gene, called an allele, represents an index of the feature level instantiated by a feature. PFCD calls for many candidates (fragments of chromosome), exhibiting a type of composition (AND) relationships. Likewise, each candidate (fragment of chromosome) comprises many features (genes). Nevertheless, each feature (gene) can assume one and only one out of many possible feature levels (alleles), suggesting an exclusive all (XOR) instantiation.

The integer format is adopted for representing multiple choices. Given K module candidates to be selected for a product family, and for each module candidate $k \mid \forall k \in [1,...,K]$, there are T_k features to be selected. Thus, a generic string of the chromosome is defined to be composed of K substrings, containing a total of $\sum_{k=1}^{K} T_k$ genes, with each substring consisting of T_k genes.

Further we introduce an allele that equals to 0 as the default value for every gene. This indicates that the corresponding feature is not contained in a module candidate. Then with Q_t possible levels for a feature, f_{kt}, the corresponding gene may assume an allele from the set, $\{0, 1, \cdots, Q_t\}$, meaning that a total number of $Q_t + 1$ alleles are available for each gene. In this way, a generic encoding enables a unified structure through which various module candidates consisting of different numbers of feature levels can be represented within a generic product family.

6.3.4.2. Hybrid constraint handling

One challenge of GAs for solving combinatorial optimization problems is constraint handling, in which genetic operators tend to manipulate the chromosomes randomly and often yield infeasible offspring. Several techniques have been proposed to handle constraints with genetic algorithms. The available techniques can be classified into four categories: the rejecting, repairing, modifying and penalty strategies. Since PFCD involves both compatibility and selection constraints, it is difficult to use a single strategy to deal with distinct characteristics of these two types of constraints simultaneously. Selection constraints refer to range restrictions, that is, feasible solutions should be within a specific range (e.g., binary restriction on decision variables). Compatibility constraints deal with restrictions on combinations. For example, when two candidates are incompatible in terms of functional features or design parameters, only one of them can be chosen for a configuration design alternative.

As a result, a hybrid constraint-handling strategy is proposed to deal with this difficulty. A hybrid strategy applies rejecting, penalty strategies, and modifying the genetic operator strategy to handle different constraints along the entire evolutionary process, as shown in Figure 6-16. At the initialization stage, a rejecting strategy is conducted to handle infeasible chromosomes. A separate constraint check module is designed as a filter. The compatibility constraints are described as a set of “IF-THEN” rules and stored in a pool. Whenever a new chromosome is initialized, it must be checked against the pool. Those chromosomes that do not satisfy certain compatibility constraints are rejected right away. In this way, only those valid chromosomes are kept in the population. A penalty strategy is only implemented at the evaluation stage where infeasible chromosomes are penalized for violating certain selection constraints. The penalty technique is used here to keep a certain amount of infeasible solutions in each generation. It does not simply reject the infeasible solutions in each generation because some of them may contain much more useful information about the optimal solutions than some feasible solutions. The penalty strategy helps acquire a balance between information preservation and selective power.

Moreover, a modifying genetic operator strategy is proposed to convert the chromosome representation scheme and generate a specialized crossover operator to maintain the feasibility of chromosomes in terms of compatibility constraints. Motivated by the design attribute encapsulation method (Qiu et al., 2002), this research proposes a Module Encapsulation Method (MEM) to modify the genetic operator. Based on the MEM, the overall modules are encapsulated into several groups, such that those modules whose candidates' combinations will result in infeasible chromosomes are encapsulated in one group. Figure 6-17 shows an incompatible interrelationship between modules. According to the identified interrelationships, incompatible modules are grouped together, for example, modules 1, 4, and 6 are encapsulated in one group, and modules 3 and 5 are in another group. In turn, all combinations of those inter-group modules always produce feasible chromosomes. According to the module groups, the module partitions are mapped into the chromosome representation scheme. Subsequently, crossover can be performed in a particular way – the encapsulated modules within a group will be handled as a whole, and the cutting points can occur only at the boundary of groups. As a result, the MEM enables the genetic operator to always generate feasible offspring, thus improving the efficiency of producing feasible chromosomes.

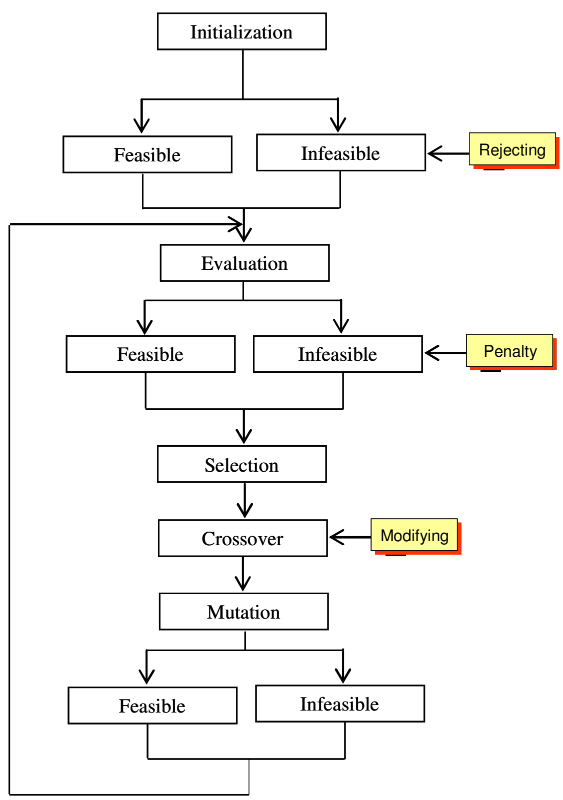


Figure 6-16 Constraint-handling mechanism of GGA

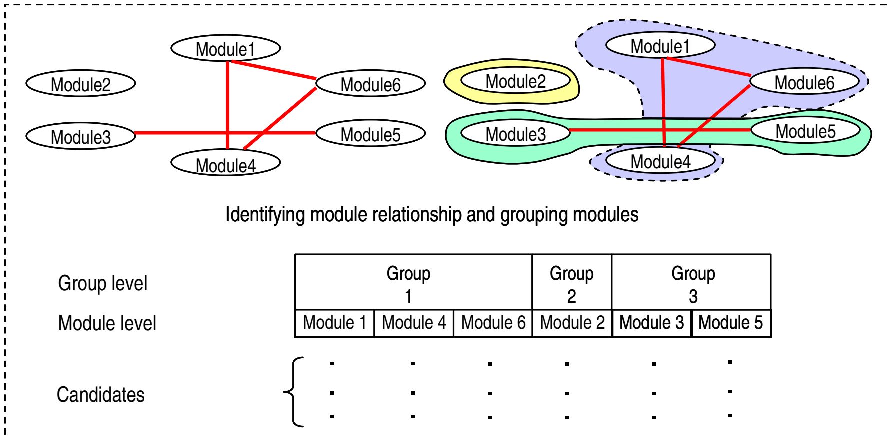


Figure 6-17 Generic operator based on encapsulation of modules

The overall procedure of the GGA is like that of the heuristic GA discussed in Chapter 5.

6.3.5. Case Study

The GGA has been applied to a type of motor family configuration design in an electronics company that produces a large variety of vibration motors for major world-leading mobile phone manufacturers. A customized electronic motor comprises mainly six modules, namely “armature”, “frame”, “bracket”, “weight”, “magnet”, and “rubber holder”. The mechanical structure of a motor is shown in Figure 6-18. Each of the six modules possesses a few candidates. Customers may ask for customized products by selecting candidates according to their preferences and needs. Each selected candidate assumes a combination of diverse feature levels and correspondingly a group of design parameters that fulfill the target functionality.

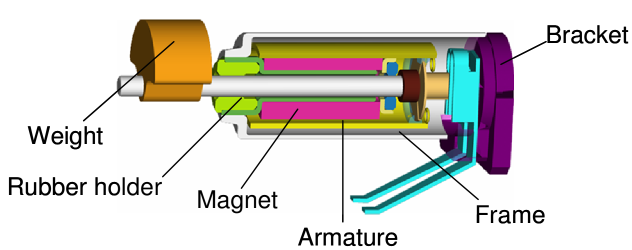


Figure 6-18 The mechanical structure of a motor

6.3.5.1. Configuration space construction

The first step for motor family configuration design is to build a generic variety structure for capturing diverse customer requirements and creating a configuration space. Figure 6-19 shows the generic variety structure of these motors, where all of the compositions and their relationships are presented as a hierarchy. Figure 6-19 also shows the compatibility constraints between feature "speed" and feature "current". A set of requirements from a particular customer are shown in Table 6-14. Based on the predefined modules and the particular customer requirements, available candidates of each module are generated to create a configuration space, as shown in Table 6-15.

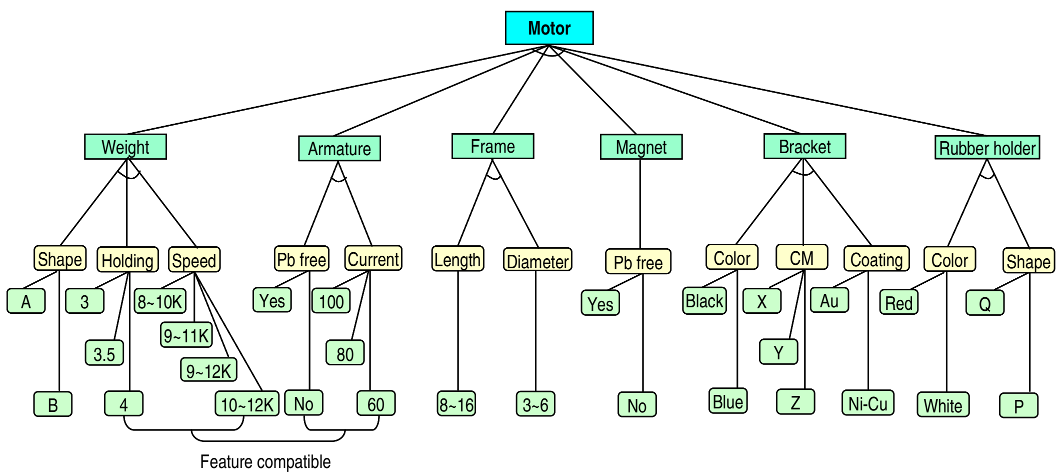


Figure 6-19 The generic variety structure of motors

Table 6-14 A particular customer's requirements

Module	Customizable function feature	Required feature level
Armature	Pb free	Yes
Frame	Length	10.5mm
	Diameter	Nil.
Bracket	Connecting method	X
	Coating	Nil.
Weight	Speed	(9000~12000)rpm
Magnet	Pb free	Nil.
Rubber holder	Shape	P

Table 6-15 Available candidates of each module

Module	Required feature level	Candidate code	Feature combination	Design parameter
Armature	Pb free = Yes	A11	Current = 100mA Pb free = Yes	Wire diameter = 50μm DimensionA = 3.8mm DimensionB = 14.5mm
		A12	Current = 80mA Pb free = Yes	Wire diameter = 35μm DimensionA = 3.5mm DimensionB = 12.5mm
		A13	Current = 60mA Pb free = Yes	Wire diameter = 25μm DimensionA = 3mm DimensionB = 11.5mm
Frame	Length = 10.5mm	A21	Length = 10.5mm Diameter = 8.5mm	Length = 10.5mm Diameter = 8.5mm
		A22	Length = 10.5mm Diameter = 13mm	Length = 10.5mm Diameter = 13mm
		A23	Length = 10.5mm Diameter = 15.5mm	Length = 10.5mm Diameter = 15.5mm
Bracket	Connecting method (CM) = X	A31	Color = Black CM = X Coating = Au	Angle = 30°
		A32	Color = Black CM = X Coating = Ni-Cu alloy	Angle = 30°
		A33	Color = Blue CM = X Coating = Au	Angle = 30°
		A34	Color = Blue CM = X Coating = Ni-Cu alloy	Angle = 30°
Weight	Speed = (9000~12000)rpm	A41	Shape = A HS = Min3kg Speed = (9000~12000)rpm	Radius = 2.5mm Length = 3mm Weight = 4.5gram Wire diameter = 50μm
		A42	Shape = A HS = Min3.5kg Speed = (9000~12000)rpm	Radius = 3.5mm Length = 3.5mm Weight = 5.5gram Wire diameter = 35μm
		A43	Shape = A HS = Min4kg Speed = (9000~12000)rpm	Radius = 4mm Length = 4.5mm Weight = 6.5gram Wire diameter = 25μm
		A44	Shape = B HS = Min3kg Speed = (9000~12000)rpm	Radius = 2mm Length = 2.5mm Weight = 4.5gram Wire diameter = 50μm
		A45	Shape = B HS = Min3.5kg Speed = (9000~12000)rpm	Radius = 3mm Length = 3mm Weight = 5.5gram Wire diameter = 35μm
		A46	Shape = B HS = Min4kg Speed = (9000~12000)rpm	Radius = 3.5mm Length = 4mm Weight = 6.5gram Wire diameter = 25μm
Magnet	Nil.	A51	Pb free = Yes	Nil.
		A52	Pb free = No	Nil.
Rubber holder	Shape = P	A61	Color = red Shape = P	Hardness = 60HB
		A62	Color = white Shape = P	Hardness = 70HB

6.3.5.2. Customer-perceived benefit and engineering costs

The particular customer-perceived benefit from a motor product and the engineering costs are measured as discussed in Chapter 5. The part-worth utilities and the part-worth standard times for all feature levels are shown in Table 6-16.

Table 6-16 Part-worth utilities and part-worth standard times

Feature level	Part-worth utility	Part-worth standard time		Feature level	Part-worth utility	Part-worth standard time	
		μ^t (second)	σ^t (second)			μ^t (second)	σ^t (second)
F111	1.85	0.65	0.01	F321	1.67	1.97	0.035
F112	1.21	0.97	0.021	F411	3.12	3.35	0.016
F121	1.12	1.45	0.21	F412	1.46	2.23	0.36
F122	1.76	0.58	0.033	F511	1.35	1.06	0.43
F123	2.65	1.42	0.31	F512	0.63	1.27	0.39
F131	1.53	0.78	0.11	F521	1.36	0.27	0.045
F132	0.87	0.21	0.03	F522	1.32	0.46	0.026
F133	0.5	0.2	0.012	F523	0.97	2.21	0.53
F134	0.52	0.18	0.023	F531	0.8	0.72	0.22
F211	2.49	1.18	0.2	F532	1.6	1.08	0.087
F212	2.32	0.19	0.013	F611	0.6	0.87	0.031
F221	1.22	1.03	0.021	F612	1.2	1.53	0.058
F222	0.65	0.62	0.008	F621	1.1	1.22	0.11
F223	0.32	0.25	0.12	F622	0.56	2.37	0.65
F311	2.18	2.3	0.02				

6.3.5.3. Generic GA solution and results

The GGA procedure is applied to search for a maximum design performance, namely, shared surplus. The chromosome string comprises 13 genes. According to the compatibility constraints shown in Figure 6-19, genes 3 and 5 are grouped together. Then the chromosome is represented as $v_k = [x_1, x_2, \{x_3, x_5\}, \ldots, x_{12}, x_{13}]$.

The result of the GGA solution is presented in Table 6-17, where the optimal result achieves the shared surplus of 0.276.

6.3.6. Efficiency Analysis

The GGA efficiency lies in generating feasible solutions efficiently and effective search along the entire GVS. This section examines the efficiency of the GGA in terms of the probability of generating feasible solutions and the GGA complexity.

Table 6-17 Optimal solution of motor family configuration design

Chromosome $v_k = [1,1,3,1,1,1,1,1,1,1,2,1,2]$		
Feature	**Feature level**	**Design parameter**
Current	100mA	Wire diameter = 50μm DimensionA = 3.8mm
Pb-free (A)	Yes	
Length	10.5mm	Length = 10.5mm Diameter = 8.5mm
Diameter	8.5mm	
Color (B)	Black	Angle = 30°
CM	X	
Coating	Ni-Cu alloy	
Shape (W)	A	Radius = 2.5mm Length = 3mm Weight = 4.5gram Wire diameter = 35μm
HS	Min3kg	
Speed	9000-12000rpm	
Pb-free (M)	No	Nil.
Color (R)	Red	Hardness = 60HB
Shape (R)	P	
Performance	0.276	

6.3.6.1. Feasible solution generation

This research adopts the MEM to modify the genetic operators. Infeasible chromosomes are encapsulated into one group, and thus combinations of the inter-group modules always produce feasible chromosomes. As a result, the probability of generating feasible solutions is improved, as proven next.

Let $A \equiv \{m_{11}^*, \cdots, m_{k2}^*, ..., m_{K3}^*\}$ be a solution. Suppose all elements of A comprise a set, $E \equiv \{e_1, \cdots, e_j, ..., e_J\}$, where J denotes the total number of elements. Encapsulate all the elements whose combinations result in infeasible solutions in the same group. That is, the set A is divided into G subsets, $S \equiv \{s_1, \cdots, s_g, ..., s_G\}$, where $G \le K$. Let $N \equiv \{n_1, \cdots, n_g, ..., n_G\}$ be a set of element number of S, where each $n_g \mid \forall g \in [1,2,...,G]$ denotes the number of elements contained in s_g. Then it is true that $J = \sum_{g=1}^{G} n_g$. Let $W \equiv \{w_1, \cdots, w_g, ..., w_G\}$ and $V \equiv \{v_1, \cdots, v_g, ..., v_G\}$ be two sets of S, where each $w_g \mid \forall g \in [1,2,...,G]$ indicates the number of possible element combinations contained in s_g, and each $v_g \mid \forall g \in [1,2,...,G]$

indicates the number of feasible element combinations. Thus, for each $v_g \mid \forall g \in [1,2,...,G]$, it is true that $v_g \leq w_g$.

The probability of generating feasible solutions without using the MEM, denoted as P_{N_MEM}, can be calculated as the following,

$$P_{N\_MEM} = \sum_{j=1}^{J} P(fea / e_j) \times P(e_j / wo\_c\_fea) \times P(wo\_c\_fea), \tag{42a}$$

s.t.

$$P(fea / e_j) = \frac{v_1}{w_1} \times \frac{v_2}{w_2} \times ... \times \frac{v_G}{w_G} = \prod_{g=1}^{G} (v_g / w_g), \tag{42b}$$

$$P(e_j / wo\_c\_fea) = 1 / \sum_{g=1}^{G} n_g, \tag{42c}$$

$$P(wo\_c\_fea) = \frac{v_1}{w_1} \times \frac{v_2}{w_2} \times ... \times \frac{v_G}{w_G} = \prod_{g=1}^{G} (v_g / w_g), \tag{42d}$$

where $P(fea / e_j)$ denotes the conditional probability of generating feasible solutions under the condition that e_j is chosen for mutation; $P(e_j / wo\_c\_fea)$ indicates the conditional probability of e_j to be chosen for mutation under the condition that the crossover operator generates feasible solutions without following the MEM; and $P(wo\_c\_fea)$ denotes the probability of generating feasible solutions after crossover without applying the MEM. Combining Eqs. (42a-d), P_{N_MEM} is calculated using the following,

$$P_{N\_MEM} = \sum_{j=1}^{\sum_{g=1}^{G} n_g} \left(e_j / wo\_c\_fea\right) \times \prod_{g=1}^{G} \left(v_g / w_g\right)^2 = \prod_{g=1}^{G} \left(v_g / w_g\right)^2. \tag{43}$$

Denote the probability of generating feasible solutions using the MEM as P_{MEM}, which is calculated as the following,

$$P_{MEM} = \sum_{g=1}^{G} P(fea/s_g) \times P(s_g / w\_c\_fea) \times P(w\_c\_fea), \tag{44a}$$

s.t.

$$P(fea/s_g) = \frac{v_g}{w_g}, \tag{44b}$$

$$P(s_g / w\_c\_fea) = n_g / \sum_{g=1}^{G} n_g , \tag{44c}$$

$$P(w\_c\_fea) = 1, \tag{44d}$$

where $P(fea/s_g)$ denotes the conditional probability of generating feasible solutions under the condition that s_g is chosen for mutation; $P(s_g / w\_c\_fea)$ indicates the conditional probability of s_g to be chosen for mutation under the condition that the crossover operator adopts the MEM; and $P(w\_c\_fea)$ denotes the probability of generating feasible solutions using the MEM for crossover. Abiding by the MEM, the crossover operator always generates feasible solutions, that is, $P(w\_c\_fea) = 1$. Combining Eqs. (44a-d), the result of P_{MEM} is given as the following,

$$P_{MEM} = \sum_{g=1}^{G} (v_g n_g / w_g \sum_{g=1}^{G} n_g). \tag{45}$$

Based on Eqs. (43) and (45), it can be proven that:

$$\begin{aligned} P_{MEM} &= \left[(v_1 w_2 \times ... w_G n_1 + v_2 w_1 ... \times w_G n_2 + ... + v_G w_1 \times ... w_{G-1} n_G) / \prod_{g=1}^{G} w_g \sum_{g=1}^{G} n_g \right] \\ &\geq \left[(v_1 \times v_2 \times ... v_G \times n_1 + v_2 v_1 ... \times v_G \times n_2 + ... + v_G v_1 \times ... v_{G-1} \times n_G) / \prod_{g=1}^{G} w_g \sum_{g=1}^{G} n_g \right] \\ &= \prod_{g=1}^{G} (v_g / w_g) \geq \prod_{g=1}^{G} (v_g / w_g)^2 = P_{N\_MEM} \end{aligned} \tag{46}$$

The result of Eq. (46) proves that the GGA does improve the probability of generating feasible solutions when adopting MEM.

6.3.6.2. Complexity analysis

Although it is always taken for granted that computers are capable of performing any computation, in practice there are a large class of programs that cannot be solved efficiently due to improper construction of the problem itself. Effective data structures thus are of primary importance for reducing complexity of the problem. Based on the GVS, the GGA constructs a configuration space represented by an AND/OR tree structure. The single formalism enables the efficient and effective search patterns, thus decreasing the difficulties in solving the PFCD problem.

With a GVS, the configuration space can be assumed to be represented as a balanced tree. Let H be the height of the tree, and n be the node number at every level of the tree. Then the total number of nodes is given as n^{H-1}. This requires $O(n^H)$ comparisons for each solution to be found. When this process continues to rank all the solutions, the complexity becomes $O(n^H)$.

Given a total number of n^{H-1} variables, a regular GA, where no generic structure is available to describe the variables, requires $O(n^{H^H})$ comparisons for each solution to be found. When this process continues to rank all the solutions, the complexity becomes $O(n^{H^H})$. Such a comparison of complexity clearly suggests that the GVS-based configuration model reduces the complexity of the GA search substantially. Therefore, the GGA is much more advantageous over a regular GA approach.

6.4. Summary

This chapter demonstrates the application of the product portfolio planning framework to customer-engineering interaction. First, an associative classification-based recommendation system is proposed for customer decision-making in an online mass customization scenario. Customers are supported to make decisions when facing overwhelming amount of information. By applying knowledge discovery techniques, the associative classification-based recommendation system overcomes the drawback of other popular methods for recommendation systems, namely content-based and collaborative-based methods. Thus, it is particularly useful in e-commerce sites that offer millions of products.

On the other hand, this chapter proposes a Kansei mining system to support affective design. By Kansei mining, the Kansei mapping patterns are generated and stored in a knowledge base and act as an interface through which the customers can interact directly with the designers. Whenever affective needs are required, the designers can start the design work without the tedious and iterative elaboration process between customers and marketing.

Finally, a generic genetic algorithm approach is developed to facilitate product family configuration design, where combinatorial explosion always occurs and is known to be mathematically intractable or NP-hard. A generic encoding scheme is developed to adapt to diverse PFCD scenarios. A hybrid constraint-handling strategy is proposed to handle complex and distinguishing constraints at different stages along the evolutionary process. The three applications are validated by the mobile phone recommendation system prototype, and case studies of mobile phone affective design and motor family configuration design, respectively.

CHAPTER 7
CONCLUSIONS AND FUTURE WORK

This concluding chapter summarizes the findings of this study in Section 7.1 and the contributions of the thesis work in Section 7.2. The limitations and possible improvements of this research are also discussed, along with avenues for future research, in Sections 7.3 and 7.4, respectively.

7.1. Conclusions

The competitive paradigm has shifted from designer-centered to customer-driven. Enterprises in all branches of industry are being forced to react to the growing individual demand. The manufacturing companies intend to provide product variety by expanding their product lines and differentiating their products, thus making their products more attractive. However, as variety keeps increasing, companies with expending products face problems of increasing costs due to an exponential growth of complexity, the inhibition of benefits from economy of scale, and exacerbation of inventory imbalances. Moreover, the practice of giving customers more choices than they actually want may lead to a paradox of mass confusion. As a result, a company must optimize its external variety with respect to the internal complexity resulting from product differentiation. Therefore, rather than creating various products in accordance with all anticipating customer needs, it becomes an important campaign for the manufacturing companies to offer the "right" product variety to the target market.

The economic success of providing a variety of product offerings depends on the ability to capture customer needs in the target market while leveraging upon customer and engineering interaction. Previous research work has emphasized customer requirement elicitation, but it is limited to the discovery of the "voice of customers" without explicitly distinguishing market and customer preference from engineering concerns. This leads to the inability to examine the combined effects of multiple product offerings on both customer satisfaction and engineering implications.

Product portfolio planning lends itself as an important strategy for portfolio decisions. It involves two main stages, namely product portfolio identification and optimization. The methodology of product portfolio identification is based on the mining of association rules so as to provide an integration of requirement information from both customer and design viewpoints within a coherent framework. For most variant product designs, where market segments have been established and product platforms have been installed, the association rule mining methodology can improve the efficiency and quality of portfolio identification by alleviating the tedious, ambiguous and error-prone process of requirement analysis enacted among customers, marketing, and designers. Generating the portfolio based on knowledge discovery from past data helps maintain the integrity of existing product and process platforms, as well as the continuity of the infrastructure and core competencies, hence leveraging existing design and manufacturing investments. The application of data mining opens opportunities for incorporating experts' experiences into the projection of portfolio patterns from historical data, thereby enhancing the ability to explore and utilize domain knowledge more effectively.

Product portfolio optimization addresses both diverse customer preferences across market segments, and engineering costs that vary with the composition of a product portfolio. By integrating marketing inputs with detailed cost information attained through coordinated product and process platforms, product portfolio optimization captures the tradeoffs between the benefits derived from providing variety to the marketplace, and the cost savings that can be realized by selecting a mix of products that can be produced efficiently within a company's manufacturing capabilities.

7.2. Contributions

The major contribution of the thesis work manifests itself through the development of a coherent framework of product portfolio planning for product portfolio decisions while leveraging both customer and engineering concerns. The deliverables are entailed from the strategy, fundamentals, methodology, tools, applications, and validation aspects, as elaborated next.

(1) At the strategy level, the following consensuses are clarified (Chapters 1 and 2):

- Distinguish functional variety from technical variety in the respective customer and functional domains; and
- Examine the importance of front-end issues with respect to the entire spectrum of platform-based product development and product family design.

(2) At the fundamental level, the following findings are achieved (Chapter 3):

- Analyze the fundamentals of product portfolio planning, which is concerned with product portfolio identification and product portfolio optimization; and
- Identify key technical challenges associated with product portfolio identification and optimization and accordingly develop the solution strategies.

(3) In terms of the methodology, the following developments are delivered (Chapters 4 and 5):

- Discover the underlying interrelationships between customer requirements and product performances, where customers' preferences are distinguished from those of engineering; and
- Model the combined effects of multiple product offerings on customer satisfaction and engineering implications.

(4) In terms of supporting tools for product portfolio planning, the following aspects are investigated:

- Apply data mining techniques to customer requirement elicitation (Chapter 4);
- Explore market research techniques for customer satisfaction modeling and customer behavior analysis in a mass customization scenario (Chapter 5); and
- Synthesize optimization techniques to deal with a number of conflicting goals from the customer and engineering perspectives regarding product portfolio optimization (Chapter 5).

(5) In terms of application, the potential of the product portfolio planning framework is demonstrated through the following (Chapter 6):

- Develop an associative classification-based recommendation system to support customer decision making in mass customization;
- Develop a Kansei mining system for customer perception modeling and affective design support; and
- Extend the product portfolio optimization framework to deal with product family configuration design.

(6) As for validation, three industrial cases are investigated, including the following:

- A case study and sensitivity analysis for generating the vibration motor portfolio in order to validate the feasibility of the product portfolio identification framework (Chapter 4);
- A case study and sensitivity analysis of notebook computer portfolio optimization to illustrate the feasibility and potential of the product portfolio optimization framework (Chapter 5);
- A prototype of mobile phone recommendation system for supporting customer decision making in online mass customization (Chapter 6);
- A case study of mobile phone affective design to justify the applicability of the Kansei mining methodology (Chapter 6); and
- A case study and efficiency analysis of motor product family configuration design to indicate the feasibility of the generic genetic algorithm approach and the shared surplus-based product family design configuration (Chapter 6).

7.3. Limitations

Product portfolio planning aims at developing decision support for manufacturing companies to offer the "right" products to match diverse customer needs. The problem formulation, system framework, architecture, and the corresponding implementations have been proposed and investigated in the thesis work. The limitations of current work mainly stem from the assumptions related to the product portfolio planning framework, in particular related to the following aspects.

(1) The reliability and effectiveness of product portfolio identification depend on the quality of knowledge. Product portfolio identification aims at reusing knowledge from historical data to facilitate the handling of requirement information and tradeoffs among

many customer, marketing and engineering concerns. The opportunity lies in taking the advantage of the wealth of customer requirement information accumulated in existing products and company databases. As a result, the performance relies on the knowledge acquired and represented. In this respect, the quality of latent knowledge plays an important role in product portfolio identification. In addition, knowledge must be constantly refined and updated to keep the customer requirement information current and valid.

(2) The robustness of product portfolio optimization needs to be improved by investigating more complex competitive scenarios. This work assumes that in the short term, competitors do not react by introducing new products or adjusting their product price. As a result, competitive reactions are implicitly modeled in the customer utilities, which are supposed to be influenced by the attributes and prices of competing products. To adapt to complex market situations, it is necessary to investigate more complex competitive scenarios.

7.4. Future Work

Product portfolio planning tackles the front-end issues of product family development. It can be enhanced by considering more complex scenarios. From a holistic view, there is still much to be desired in order to achieve system-wide solutions for product family design and platform-based product development. In this regard, the following areas appear to be promising avenues for further research efforts.

(1) Active competition modeling. One of the fruitful directions would be the modeling of active competition. In most cases, complete information about the competitors is not available. To maintain dominance, competitors always adjust their competition strategies. For example, most competitors eventually react to new entries with changes in their prices. The dynamic markets and uncertain information make it difficult to make decisions.

Therefore, the competitive scenarios and market dynamics should be analyzed to develop a systematic approach to study decision making in conflicting situations where two or more decision makers are involved. This may be verified by explicitly modeling competitive reactions within a game theoretic framework or by deriving competitive strategies in conjoint analysis under the Nash equilibrium concept (see e.g., Choi and DeSarbo, 1994).

(2) Dynamic customer behavior analysis. If interactions among diverse product attributes are to be considered, customer behavior will become more dynamic and complicated. The interactions may be verified by factor analysis. Extended utility functions, such as a quadratic utility function, afford the opportunity for dynamic customer behavior analysis. The extended utility function may be constructed using central composite designs, which contains an imbedded fractional factorial design with center points that allow the estimation of curvature and second-order effects. Another area of interest would be discrete choice analysis for predicting the choices that customers will make between alternatives provided by a product portfolio. It encompasses a variety of experimental design techniques, data collection procedures, and statistical procedures (see e.g., Watson, et al., 2002; Hayakawa, 1976).

(3) Product family design support. Although the basic principles of product family design are understood and well documented in the literature, quite a few fundamental issues require further examination, for instance, to what extent can a product family architecture and platform best represent the capability of an enterprise? How can product families be matched with an existing set of resources and enterprise capabilities? How should product platforms and architectures evolve in accordance with changes in customers' requirements, product technologies and enterprise capabilities? Product architecture and platform modeling

is one of the fruitful research topics. Comparing with numerous efforts in product family optimization design, this field has so far received least attention, and little achievement has been reported. It is imperative to call for rigorous research that synthesizes useful ingredients from those establishments in the artificial intelligence field such as configuration topology, software product families, and architectural modeling (see e.g., Jiao, et al., 2006).

(4) Product platform risk management. The risks related to product family development need to be addressed properly. Developing product platforms in most cases requires more investments and development time than developing a single product, which may delay the time to market and affect the return on investment time. The risks may undermine the competitiveness of the entire product line, and therefore a broad array of products may feel the pain. Organizational forces may also hinder the ability to balance commonality and distinctiveness (see e.g., Meyer and Lehnerd, 1997; Robertson and Ulrich, 1998).

(5) Extended platforms for collaborative product families. A product family should ideally be built on sharing a multidimensional core of assets such as standardized components, manufacturing, supply and distribution processes, customer segmentation and brand positioning. To support the coordination of the demand and supply chains with product families, it is necessary to extend platform thinking to the entire continuum of product fulfillment, including customer platforms, brand platforms, product platforms, process platforms, and global platforms. Greater complexity must be introduced to product family design decisions when considering more decision variables or parameters pertinent to the coordination across the product, manufacturing process and supply chain domains.

References

Adamo, J.-M., 2001, *Data Mining For Association Rules And Sequential Patterns: Sequential And Parallel Algorithms*, New York: Springer.

Agard, B., and Kusiak, A., 2004, Data-mining-based methodology for the design of product families, *International Journal of Production Research*, 42(15): 2955-2969.

Aggarwal, C.C., and Yu, P.S., 1998, A new framework for itemset generation, Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, pp. 18-24, Seattle, WA.

Agrawal, R., Imielinski, T., and Swami, A., 1993, Mining association rules between sets of items in massive database, Proceedings of the ACM/SIGMOD International Conference on Management of Data, pp. 207-216, Washington, DC.

Agrawal, R., and Srikant, R., 1994, Fast algorithms for mining association rules in large databases, Proceedings of 20th International Conference on Very Large Data Bases, pp. 487-499, Santiago de Chile, Chile.

Akundi, S., Simpson, T.W., and Reed, P.M., 2005, Multi-objective design optimization for product platform and product family design using genetic algorithms, ASME Design Engineering Technical Conferences, DETC2005/DAC-84905, Long Beach, CA.

Allen, K.R., and Carlson-Skalak, 1998, Defining product architecture during conceptual design, ASME Design Engineering Technical Conference, DETC98/DTM-5650, Atlanta, GA.

Ardissono, L, Felfernig, A., Friedrich, G., Goy, A., Jannach, D., Petrone, G., Schäfer, R., and Zanker, M., 2003, A framework for the development of personalized, distributed web-based configuration systems, *AI Magazine*, 24(3): 93-110.

Arrow, K.J., and Raynaud, H., 1986, *Social Choice and Multicriterion Decision-Making*, The MIT Press, Cambridge, MA.

Avery, C., and Zeckhauser, R., 1997, Recommender systems for evaluating computer messages, *Communications of the ACM*, 40(3): 88-89.

Aydiny, A.O., and Gugor, A., 2005, Effective relational database approach to represent bills-of-materials, *International Journal of Production Research*, 43(6): 1143-1170.

Balakrishnan, P.V.S., and Jacob, V.S., 1996, Genetic algorithms for product design, *Management Science*, 42(1): 1105-1117.

Baldwin, C.Y., and Clark, K.B., 2000, *Design Rules: The Power of Modularity*, MIT Press, Cambridge, MA.

Basu, C., Hirsh, H., and Cohen, W., 1998, Recommendation as classification: using social and content-based information in recommendation, Proceedings of the 1998 Workshop on Recommender Systems, pp. 11-15, Menlo Park, CA.

Bei, Y., and MacCallum, K.J., 1995, Decision support for configuration management in product development, Proceedings of the 3rd International Conference on Computer Integrated Manufacturing, Vol. 1, pp. 278-286, World Scientific, Singapore.

Ben-Akiva, M., and Lerman, S., 1985, *Discrete Choice Analysis: Theory and Application to Travel Demand*, The MIT Press, Cambridge, MA.

Berson, A., Smith, S., and Thearling, K., 1999, *Building Data Mining Applications for CRM*, McGraw-Hill, New York.

Bi, Z.M., and Zhang, W.J., 2001, Modularity technology in manufacturing: taxonomy and issues, *International Journal of Advanced Manufacturing Technology*, 18(5): 381-390.

Blecker, T., and Kreutler, G., 2004, An advisory system for customers' objective needs elicitation in mass customization, Proceedings of the 4th International ICSC Symposium on Engineering Of Intelligent Systems, pp. 3-13, Portugal.

Bohm, M.R., and Stone, R.B., 2004, Representing functionality to support reuse: conceptual and supporting functions, ASME Design Engineering Technical Conferences, DETC2004-57693, Salt Lake City, Utah.

Bramham, J., and MacCarthy, B., 2003, Matching configurator attributes to business strategy, The 2nd World Congress on Mass Customization and Personalization, CD-ROM, Munich.

Brin, S., Motwani, R., Ullman, J.D., and Tsur, S., 1997, Dynamic itemset counting and implication rules for market basket data, Proceedings of the ACM SIGMOD Conference on Management of Data, pp. 255-264, Montreal, Canada.

Byrne, J.G., and Barlow, T., 1993, Structured brainstorming: a method for collecting user requirements, Proceedings of the 37th Annual Meeting of the Human Factors and Ergonomics Society, pp. 427-431, Seattle, WA.

Carbonell, J.G., 1992, Natural Language Understanding, In *Encyclopedia of Artificial Intelligence*, Shapiro, S.C. (Ed.), pp. 997-1015, John Wiley and Sons, Inc., NY.

Chandler, C., and Williams, M., 1993, Strategic shift: A slump in car sales forces Nissan to start cutting swollen costs, *Wall Street Journal*, New York, NY, A1.

ChangChien, S.W., and Lu, T.C., 2001, Mining association rules procedure to support on-line recommendation by customers and products fragmentation, *Expert Systems with Applications*, 20(4): 325-335.

Chen, C.-H., Khoo, L.P., and Yan, W., 2000, An investigation to the elicitation of customer requirements using sorting techniques and fuzzy evaluation, Proceedings of the 6th Asia Pacific Management Conference, pp. 45-55, Tainan, Taiwan.

Chen, C.-H., Khoo, L.P., and Yan, W., 2002, A strategy for acquiring customer requirement patterns using laddering technique and ART2 neural network, *Advanced Engineering Informatics*, 16(3): 229-240.

Chen, C.-H., and Occeña, L.G., 1995, An expert system for wood head golf clubs design, Proceedings of the 4th Industrial Engineering Research Conference, pp. 926-932, Nashville, TN, USA.

Chen, K.D., and Hausman, W.H., 2000, Technical note: Mathematical properties of the optimal product line selection problem using choice-based conjoint analysis, *Management Science*, 46(2): 327-332.

Chen, M., Han, J., and Yu, P., 1996, Data mining: An overview from database perspective, *IEEE Transactions on Knowledge and Data Engineering*, 8(6): 866-883.

Cheung, C.F., Lee, W.B., Wang, W.M., Chu, K.F., and To, S., 2003, A multi-perspective knowledge-based system for customer service management, *Expert Systems with Applications*, 24(3): 457-470.

Chidambaram, B., and Agogino, A.M., 1999, Catalog-based customization, Proceedings of ASME Design Engineering Technical Conferences, DETC99/DAC-8675, Las Vegas, Nevada, USA.

Child, P., Diederichs, R., Sanders, F.H., and Wisniowski, S., 1991, SMR forum: The management of complexity, *Sloan Management Review*, 33(1): 73-80.

Choi, S.C., and DeSarbo, W.S., 1994, A conjoint-based product designing procedure incorporating price competition, *Journal of Product Innovation Management*, 11(5): 451-459.

Christopher, M.C., McDonald, M., and Wills, G., 1980, *Introducing Marketing*, Pan, London.

Clausing, D., 1994, *Total Quality Development: A Step-by-Step Guide to World Class Concurrent Engineering*, ASME Press, New York.

Collier, D.A., 1981, The measurement and operating benefits of component part commonality, *Decision Sciences*, 12(1): 85-96.

Cooper, R., Edgett, S., and Kleinschmidt, E., 2001, Portfolio management for new product development: Results of an industry practices study, *R & D Management*, 31(4): 361-381.

Cortazzi, D., and Roote, S., 1975, *Illuminative Incident Analysis*, McGraw-Hill, London.

Costa, C.A., and Young, R.I.M., 2001, Product range models supporting design knowledge reuse, Proceedings of IMechE, Part B, *Journal of Engineering Manufacture*, 215(3): 323-337.

Cox, M.W., and Alm, R., 1998, The right stuff-America's move to mass customization, Annual Report, Federal Reserve Bank of Dallas.

Cross, N., 2000, *Engineering Design Methods: Strategies for Product Design* (3rd Edition), John Wiley & Sons Ltd, Chichester, UK.

Cutherell, D., 1996, Product architecture, *The PDMA Handbook of New Product Development*, Rosenau, M., Griffin, A., Castellion, G., Anschuetz, N. (Eds.), John Wiley & sons Ltd, Chichester, UK.

Da Silveira, G., Borenstein, D., and Fogliatto, F.S., 2001, Mass customization: Literature review and research directions, *International Journal of Production Economics*, 72(1): 1-13.

Dai, Z., and Scott, M.J., 2004, Product platform design through sensitivity analysis and cluster analysis, ASME Design Engineering Technical Conferences, DETC2004/DAC-57464, Salt Lake City, UT.

de Groote, X., 1994, Flexibility and product variety in lot-sizing models, *European Journal of Operational Research*, 75(2): 264-274.

De Lit, P.G., and Delchambre, A., 2003, *Integrated Design of a Product Family and Its Assembly System*, Kluwer Academic Publishers, MA.

De Weck, O.L., Suh, E.S., and Chang, D., 2003, Product family and platform portfolio optimization, ASME Design Engineering Technical Conferences, DETC03/DAC-48721, Chicago, IL.

Demirbilek, O., and Sener, B., 2003, Product design, semantics and emotional response, *Ergonomics,* 46(13/14): 1346-1360.

Dobrescu, G., and Reich, Y., 2003, Progressive sharing of modules among product variants, *Computer-Aided Design*, 35(9): 791-806.

Dobson, G., and Kalish, S., 1988, Positioning and pricing a product line, *Marketing Science*, 7(2): 107-125.

Dobson, G., and Kalish, S., 1993, Heuristics for pricing and positioning a product-line using conjoint and cost data, *Management Science*, 39(2): 160-175.

Dobson, G., and Yano, C.A., 1994, Product line and technology selection with shared manufacturing and engineering design resources, Paper provided by Rochester, Business -

Center for Manufacturing and Operations Management in its paper series with number (RePEc:fth:robuma:95-01), http://ideas.repec.org/p/fth/robuma/95-01.html.

Du, X., Jiao, J., and Tseng, M.M., 2001, Architecture of product family: Fundamentals and methodology, *Concurrent Engineering: Research and Application*, 9(4): 309-325.

Du, X., Jiao, J., and Tseng, M.M., 2003, Identifying customer need patterns for customization and personalization, *Integrated Manufacturing Systems*, 14(5): 387-396.

Duray, R., and Milligan, G.W., 1999, Improving customer satisfaction through mass customization, *Quality Progress*, 32(8): 60-66.

Duray, R., Ward, P.T., Milligan, G.W., and Berry, W.L., 2000, Approaches to mass customization: Configurations and empirical validation, *Journal of Operations Management*, 18(6): 605-625.

Economist, 2001, Business special: A long march mass customization, *The Economist*, 360(8230): 63-65.

Eppinger, S.D., Whitney, D.E., Smith, R.P., and Gebala, D.A., 1994, A model-based method for organizing tasks in product development, *Research in Engineering Design*, 6(1): 1-13.

Erens, F., and Verhulst, K., 1997, Architectures for product families, *Computers in Industry*, 33(2-3): 165-178.

Erixon, G., and Ostgren, B., 1993, Synthesis and evaluation tool for modular designs, Proceedings of the International Conference on Engineering Design, pp. 898-905, Hague, Netherlands.

Erlandsson, A., Erixon, G., and Ostgren, B., 1992, Product modules - the link between QFD and DFA? Proceedings of the International Forum on Product Design for Manufacture and Assembly, pp. 523-538, Newport, RI.

Felfernig, A., Friedrich, G., and Jannach, D., 2001, Conceptual modeling for configuration of mass customizable products, *Artificial Intelligence in Engineering*, 15(2): 165-176.

Fellini, R., Kokkolaras, M., Papalambros, P.Y., and Perez-Duarte, A., 2002, Platform selection under performance loss constraints in optimal design of product families, ASME Design Engineering Technical Conferences, DETC02/DAC-34099, Montreal, Canada.

Finch, W., 1999, Set-based models of product platform design and manufacturing processes, ASME Design Engineering Technical Conferences, DETC99/DTM-8763, Las Vegas, NV.

Fisher, M., Ramdas, K., and Ulrich, K., 1999, Component sharing in the management of product variety: A study of automotive braking systems, *Management Science*, 45(3): 297-315.

Fixson, S.K., 2002, Linking modularity and cost: A methodology to assess cost implications of product architecture differences to support product design, Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA.

Fixson, S.K., 2005, Product architecture assessment: A tool to link product, process, and supply chain design decisions, *Journal of Operations Management*, 23(3-4): 345-369.

Fox, C., 1992, *Information Retrieval: Data Structures and Algorithms*, Prentice-Hall, Englewood, NJ.

Francisco, G.-S., Rafael, V.-G., and Rodrigo, M.-B., 2005, An integrated approach for developing e-commerce applications, *Expert Systems with Applications*, 28(2): 223-235.

Freitas, A.A., 2002, *Data Mining and Knowledge Discovery With Evolutionary Algorithms*, Springer, NY.

Fujita, K., Sakaguchi, H., and Akagi, S., 1999, Product variety deployment and its optimization under modular architecture and module commonalization, ASME Design Engineering Technical Conferences, DETC99/DFM-8923, Las Vegas, NV.

Fujita, K., and Yoshida, H., 2004, Product variety optimization simultaneously designing module combination and module attributes, *Concurrent Engineering: Research and Application*, 12(2): 105-118.

Fukuda, S., and Matsuura, Y., 1993, Prioritizing the customer's requirements by AHP for concurrent design, Proceedings of Design for Manufacturability, DE-Vol.52, ASME, pp. 13-19.

Fung, R.Y.K., and Popplewell, K., 1995, The analysis of customer requirements for effective rationalization of product attributes in manufacturing, Proceedings of 3rd International Conference on Manufacturing Technology, pp.287-296, HK.

Fung, R.Y.K., Popplewell, K., and Xie, J., 1998, An intelligent hybrid system for customer requirements analysis and product attribute targets determination, *International Journal of Production Research*. 36(1): 13-34.

Fung, R.Y.K., Tang, J., Tu, Y., and Wang, D., 2002, Product design resources optimization using a non-linear fuzzy quality function deployment model, *International Journal of Production Research*. 40(3): 585-599.

Gen, M., and Cheng, R., 1997, *Genetic Algorithms & Engineering Design*, John Wiley and Sons, New York.

Gen, M., and Cheng, R., 2000, *Genetic Algorithm and Engineering Optimization*, John Wiley and Sons, New York.

Gershenson, J.K., Prasad, G.J., and Zhang, Y., 2003, Product modularity: Definitions and benefits, *Journal of Engineering Design*, 14(3): 295-313.

Gonzalez-Zugasti, J.P., Otto, K., and Baker, J., 2000, A method for architecting product platforms, *Research in Engineering Design*, 12(2): 61-72.

Gonzalez-Zugasti, J.P., Otto, K., and Baker, J., 2001, Assessing value for platformed product family design, *Research in Engineering Design*, 13(1):30-41.

Green, P.E., and DeSarbo, W.S., 1978, Additive decomposition of perceptions data via conjoint analysis, *Journal of Consumer Research*, 5(1): 58-65.

Green, P.E., and Krieger, A.M., 1985, Models and heuristics for product line selection, *Marketing Science*, 4(1):1-19.

Green, P.E., and Krieger, A.M., 1989, Recent contributions to optimal product positioning and buyer segmentation, *European Journal of Operational Research*, 41(2): 127-141.

Green, P.E., and Krieger, A.M., 1996, Individualized hybrid models for conjoint analysis, *Management Science*, 42(6): 850-867.

Green, P.E., Krieger, A.M., and Wind, Y, 2001, Thirty years of conjoint analysis: Reflections and prospects, *Interface*, 31(3): S56-S73.

Griffin, A., and Hauser, J.R., 1992, The voice of the customer, *Marketing Science*, 12(1): 1-27.

Guiltinan, J.P., 1993, A strategic framework for assessing product line additions, *Journal of Product Innovation Management*, 10(2): 136-147.

Guo, F., and Gershenson, J.K., 2004, A comparison of modular product design methods on improvement and iteration, ASME Design Engineering Technical Conferences, Salt Lake City, UT.

Han, J., and Kamber, M., 2001, *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers, San Francisco, CA.

Hart, W.L.C., 1995, Mass customization: Conceptual underpinnings, opportunities and limits, *International Journal of Service Industry Management*, 6(2): 36-45.

Hassan, R., Weck, O., and Springmann, P., 2004, Architecting a communication satellite product line, The 22th AIAA International Communications Satellite Systems Conference & Exhibit, AIAA2004-3150, Monterey, CA.

Hauge, P.L., and Stauffer, L.A., 1993, ELK: A method for eliciting knowledge from customers, Design and Methodology Conference, DE-Vol.53, ASME, pp.73-81, New York.

Hayakawa, H., 1976, Consumer theory when prices and real income affect preferences, *Journal of Southern Economic*, 43(1): 840-845.

Hazelrigg, G.A., 1998, A framework for decision-based engineering design, *ASME Journal of Mechanical Design*, 120(4): 653-658.

Hegge, H.M.H., and Wortmann, J.C., 1991, Generic bill-of-material: A new product model, *International Journal of Production Economics*, 23(1-3): 117-128.

Helander, M.G., and Khalid, H.M., 2005, Affective and pleasurable design, *Handbook of Human Factors and Ergonomics*, Salvendy, G. (Ed.), Third Edition, Wiley Interscience, NY.

Helander, M.G., Khalid, H.M., and Tham, M.P., 2001, Proceedings of the International Conference on Affective Human Factors Design, ASEAN Academic Press, London.

Helander, M.G., and Tham, M.P., 2003, Hedonomics – Affective human factors design, *Ergonomics,* 46(13/14): 1269-1272.

Henderson, B.D., 1970, *The Product Portfolio*, Boston Consulting Group, Boston, MA.

Henderson, R.M., and Clark, K.B., 1990, Architecture innovation: The reconfiguration of existing product technologies and the failure of established firms, *Administrative Science Quarterly,* 35(1): 9-30.

Hill, W., Stead, L., Rosenstein, M., and Furnas, G., 1995, Recommending and evaluating choices in a virtual community of use, Proceedings of the 1995 ACM Conference on Factors in Computing Systems, pp. 194-201, New York, USA.

Hillier, M.S., 2000, Component commonality in multiple-period, assembly-to-order systems, *IIE Transactions*, 32(8): 755-766.

Ho, T.H., and Tang, C.S., 1998, *Product Variety Management: Research Advances*, Kluwer Academic Publishers, Boston, MA.

Holland, J.H., 1992, *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, MA.

Hölttä, K., and Salonen, M., 2003, Comparing three modularity methods, ASME Design Engineering Technical Conferences, DETC2003/DTM-48649, Chicago, IL.

Hölttä, K., Tang, V., and Seering, W., 2003, Modularizing product architectures using dendrograms, International Conference on Engineering Design, Stockholm.

Hotelling, H.H., 1929, Stability in competition, *Economic Journal*, 39(1): 47-51.

Huang, C.-C., and Liang, W.Y., 2001, A formalism for designing with modules, *Journal of the Chinese Institute of Industrial Engineers*, 18(3): 13-20.

Huang, G.Q., and Mak, K.L., 2000, Webid: A Web-based framework to support early supplier involvement in new product development, *Robotics and Computer Integrated Manufacturing*, 16(2-3): 169-179.

Huffman, C., and Kahn, B., 1998, Variety for sale: Mass customization or mass confusion? *Journal of Retailing*, 74(4): 491-513.

Hvam, L., 2004, A multi-perspective approach for the design of product configuration systems – an evaluation of industry applications, International Conference on Economic, Technical and Organizational Aspects of Product Configuration Systems, Technical University of Denmark, Lyngby, Denmark.

Ishii, K., and Martin, M.V., 1996, Design for variety: A methodology for understanding the costs of product proliferation, Proceedings of 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference, pp. 18-22, Irvine, CA.

Jaime, M., 1998, Product line streamlining: A methodology to guide product costing and decision making, M.S. Thesis, Massachusetts Institute of Technology, Cambridge, MA.

Jenkins, S., 1995, Modeling a perfect profile, *Marketing*, July, pp. 6.

Jiao, J., Kumar, A., Lim, C.M., and Tseng, M.M., 2004a, Flexibility study of product and process platforms: a real-option approach, The 11th ISPE International Conference on Concurrent Engineering, CD-ROM, Beijing, China.

Jiao, J., Simpson, T.W., and Siddique, Z., 2006, Product family design and platform-based product development: A start-of-the-art review, *Journal of Intelligent Manufacturing*, In Print.

Jiao, J., and Tseng, M.M., 1999a, A methodology of developing product family architecture for mass customization, *Journal of Intelligent Manufacturing*, 10(1): 3-20.

Jiao, J., and Tseng, M.M., 1999b, A pragmatic approach to product costing based on standard time estimation, *International Journal of Operations & Production Management*, 19(7): 738-755.

Jiao, J., and Tseng, M.M., 2000, Fundamentals of product family architecture, *Integrated Manufacturing Systems*, 11(7): 469-483.

Jiao, J., and Tseng, M.M., 2004, Customizability analysis in design for mass customization, *Computer-Aided Design*, 36(8): 745-757.

Jiao, J., Tseng, M.M., Duffy, V.G., and Lin, F., 1998, Product family modeling for mass customization, *Computers and Industrial Engineering*, 35(3-4): 495-498.

Jiao, J., Tseng, M.M., Ma, Q., and Zou, Y., 2000, Generic bill of materials and operations for high-variety production management, *Concurrent Engineering: Research and Application*, 8(4): 297-322.

Jiao, J., Zhang, L., and Pokharel, S., 2003, Process platform planning for mass customization, The 2nd Interdisciplinary World Congress on Mass Customization and Personalization, CD-ROM Proceedings, Munich, Germany.

Jiao, J., and Zhang, Y., 2005, Product portfolio identification based on association rule mining, *Computer-Aided Design*, 37(2): 149-172.

Jiao, J., Zhang, Y., and Wang, Y., 2004b, Product portfolio planning considering customer-engineering interaction: problem formulation, The 7th International Conference on Work with Computing Systems, CD-ROM, Kuala Lumpur, Malaysia.

Jordan, P.W., 2000, The four pleasures - A framework for pleasures in design, In P.W. Jordan (Ed.), Proceedings of Conference on Pleasure Based Human Factors Design, Groningen, pp. 206-217, Philips Design, The Netherlands.

Kamrani, A.K., and Gonzalez, R., 2003, A genetic algorithm-based solution methodology for modular design, *Journal of Intelligent Manufacturing*, 14(6): 599-616.

Kano, N., Seraku, N., Takahashi, F., and Tsuji, S., 1984, Attractive and must-be quality (in Japanese), *Hinshitsu*, 14(2): 39-48.

Kaul, A., and Rao, V.R., 1995, Research for product positioning and design decisions: an integrative review, *International Journal of Research in Marketing*, 12(4): 293-320.

Keeney, R.L., and Raiffa, H., 1976, *Decision with Multiple Objectives: Preferences and Value Tradeoffs*, John Wiley & Sons, New York.

Kelly, GA., 1955, *The Psychology of Personal Constructs*, W.W. Norton, New York.

Khalid, H.M., 2001, Towards affective collaborative design, In M.J. Smith, G. Salvendy, D. Harris and R.J. Koubek (Eds.), *Usability Evaluation and Interface Design*, Vol. 1 of the Proceedings of HCI International 2001, pp. 370-374, Lawrence Erlbaum, NJ.

Khalid, H.M., 2004, Conceptualizing affective human factors design, *Theoretical Issues in Ergonomics Science*, 5(1): 1-3.

Khalid, H.M., and Helander, M.G., 2004, A framework for affective customer needs in product design, *Theoretical Issues in Ergonomics Science*, 5(1): 27-42.

Kim, K., and Chhajed, D., 2001, An experimental investigation of valuation change due to commonality in vertical product line extension, *Journal of Product Innovation Management*, 18(4): 219-230.

Kogut, B., and Kulatilaka, N., 1994, Option thinking and platform investment: investing in opportunity, *California Management Review*, 36(20): 52-71.

Kohli, R., and Krishnamurti, R., 1987, A heuristic approach to product design, *Management Science*, 33(12): 1523-1533.

Kohli, R., and Sukumar, R., 1990, Heuristics for product-line design using conjoint analysis, *Management Science*, 36(12): 1464-1478.

Kota, S., Sethuraman, K., and Miller, R., 2000, A metric for evaluating design commonality in product families, *Journal of Mechanical Design*, 122(4): 403-410.

Kotler, P., 1994, *Marketing Management*, Prentice-Hall, Englewood Cliffs, NJ.

Krishnan, V., and Ulrich, K., 2001, Product development decisions: A review of the literature, *Management Science*, 47(1): 1-21.

Kuhfeld, W.F., 2004, Conjoint analysis, SAS technical support resources, TS-689G, http://support.sas.com/techsup/technote/ts689g.pdf.

Kusiak, A., and Huang, C.C., 1996, Development of modular products, *IEEE Transactions on Components, Packaging, and Manufacturing Technology*, *Part A*, 19(4): 523-538.

LaChance-Porter, S., 1993, Impact of user focus groups on the design of new products, Proceedings of the 14th National On-line Meeting, pp.265-271, NY.

Lancaster, K., 1990, The economics of product variety: A survey, *Marketing Science*, 9(3): 189-211.

Lee, H.L., and Billington, C., 1994, Designing products and processes for postponement, in *Management of Design: Engineering and Management Perspectives*, Dasu, S., and Eastman, C. (Eds.), pp. 105-122, Kluwer Academic Publishers, Boston, MA.

Li, H., and Azarm, S., 2002, An approach for product line design selection under uncertainty and competition, *Transactions of the ASME, Journal of Mechanical Design*, 124(3): 385-392.

Liang, W.Y., and O'Grady, P., 2000, A constrained evolutionary search formalism for remote design with modules, *International Journal of Computer Integrated Manufacturing*, 13(2): 65-79.

Lilien, G.L., Kotler, P., and Moorthy, K.S., 1992, *Marketing Models*, Prentice-Hall International, Englewood Cliffs, NJ.

Lin, C.T., and Lee, C.S.G., 1996, *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*, Prentice Hall, NJ.

Liu, B., Hsu, W., and Ma, Y., 1998, Integrating classification and association rule mining, Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, pp. 80-86, New York.

Louviere, J., Anderson, D., White, J.B., and Eagle, T.C., 1990, Predicting preferences for new Product configurations: A high-tech example, Proceedings of the Conference on Modeling the Innovation: Communications, Automation and Information Systems, pp. 53-61, Rome, Italy.

MacDuffie, J.P., Sethuraman, K., and Fisher, M.L., 1996, Product variety and manufacturing performance: Evidence from the international automotive assembly plant study, *Management Science*, 42(3): 350-369.

Maiden, N.A.M., and Rugg, G., 1996, ACRE: Selection methods for requirements acquisition, *Software Engineering Journal*, 11(3): 183-192.

Malmström, J., and Malmqvist, J., 1998, Tradeoff analysis in product structures: A case study at Celsius Aerotech, Proceedings of NordDesign, pp. 187-196, Stockholm.

Mani, D.R., Drew, J., Betz, A., and Datta, P., 2001, Amalgamation of statistics and data mining techniques: Explorations in customer lifetime value modeling, In: *Knowledge*

Discovery for Business Information Systems, Abramowicz, W., Zurada, J. (Eds.), pp. 229-250, Kluwer Academic Publishers, Boston, MA.

Männistö, T., 2000, A conceptual modeling approach to product families and their evolution, Ph.D thesis, Helsinki University of Technology, Acta Polytechnica Scandinavica, Mathematics and Computing Series, No. 106, Espoo, Finland.

Manski, C., 1977, The structure of random utility model of choice, *Journal of Economics*, 3(3): 205-228.

Markus, A., and Váncza, J., 1998, Product line development with customer interaction, *CIRP Annals*, 47(1): 361-364.

Martin, M.V., and Ishii, K., 1997, Design for variety: Development of complexity indices and design charts, ASME Design Engineering Technical Conferences, DFM-4359, Sacramento, CA.

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

Martinez, M.T., Favrel, J., and Ghodous, P., 2000, Product family manufacturing plan generation and classification, *Concurrent Engineering: Research and Applications*, 8(1): 12-23.

Maupin, A.J., and Stauffer, L.A., 2000, A design tool to help small manufacturers reengineer a product family, ASME Design Engineering Technical Conferences, DETC99/DTM-14568, Baltimore, MD.

McAdams, D.A., Stone, R.B., and Wood, K.L., 1999, Functional interdependence and product similarity based on customer needs, *Research in Engineering Design*, 11(1):1-19.

McAdams, D.A., and Wood, K.L., 2002, A quantitative similarity metric for design-by-analogy, *ASME Journal of Mechanical Design*, 124(2): 173-182.

McBride, R.D., and Zufryden, F.S., 1988, An integer programming approach to the optimal product line selection problem, *Marketing Science,* 7(2): 126-140.

McGrath, M., 1995, *Product Strategy for High-Technology Companies*, Irwin Professional Publishing, New York,

McKay, A., de Pennington, A., and Baxter, J., 2001, Requirements management: A representation scheme for product, *Computer-Aided Design*, 33(7): 511-520.

Mead, M., 1928, *Coming of Age in Samoa*, William Morrow, New York.

Meyer, M., and Lehnerd, A.P., 1997, *The Power of Product Platform – Building Value and Cost Leadership*, Free Press, New York.

Meyer, M., and Utterback, J., 1993, The product family and the dynamics of core capability, *Sloan Management Review*, Spring, pp.29-47.

Michalek, J.J., Feinberg, F.M., and Papalambros, P.Y., 2005, Linking marketing and engineering product design decisions via analytical target cascading, *Journal of Product Innovation Management*, 22(1): 42-62.

Mikkola, J.H., and Gassmann, O., 2003, Managing modularity of product architectures: Towards an integrated theory, *IEEE Transactions on Engineering Management*, 50(2): 204-218.

Moffat, S., 1990, Japan's new personalized production, *Fortune*, 122(10): 132-135.

Monroe, K., Sunder, S., Wells, W.A., and Zoltners, A.A., 1976, A multi-period integer programming approach to the product mix problem, Proceedings of the American Marketing Association Meeting, Bernhardt, K. (Ed.), pp. 493-497.

Moore, W.L., Louviere, J.J., and Verma, R., 1999, Using conjoint analysis to help design product platforms, *Journal of Product Innovation Management*, 16(1): 27-39.

Morgan, L.O., Daniels, R.L., and Kouvelis, P., 2001, Marketing/Manufacturing tradeoffs in product line management, *IIE Transactions*, 33(11): 949-962.

Mosher, R., 1999, The use of a product end of life process to effectively manage a product portfolio, M.S. Thesis, Massachusetts Institute of Technology, Cambridge, MA.

Muffatto, M., and Roveda, M., 2002, Product architecture and platforms: A conceptual framework, *International Journal of Technology Management*, 24(1): 1-16.

Nagamachi, M., 1989, *Kansei Engineering*, Kaibundo Publisher, Tokyo.

Nagamachi, M., 1996, *Introduction of Kansei Engineering*, Japan Standard Association, Tokyo.

Nair, S.K., Thakur, L.S., and Wen, K., 1995, Near optimal solutions for product line design and selection: Beam search heuristics, *Management Science*, 41(5): 767-785.

Nelson, S.A. II, Parkinson, M.B., and Papalambros, P.Y., 2001, Multicriteria optimization in product platform design, *ASME Journal of Mechanical Design*, 123(2): 199-204.

Nemhauser, G.L., and Wolsey, L.A., 1988, *Integer and Combinatorial Optimization*, Wiley Publisher, New York.

Nobelius, D., and Sundgren, N., 2002, Managerial issues in parts sharing among product development projects: a case study, *Journal of Engineering Technology Management*, 19(1): 59-73.

Obitko, M., 2003, *Introduction to Genetic Algorithm*, http://labe.felk.cvut.cz/~obitko/ga/.

O'Grady, P., and Liang, W.-Y., 1998, An object oriented approach to design with modules, *Computer Integrated Manufacturing Systems*, 11(4): 267-283.

Olewnik, A., and Lewis, K., 2005, Can a house without a foundation support design? ASME Design Engineering Technical Conferences, DETC-84765, Long Beach, CA.

Olsen, K.A., and Sætre, P., 1996, Managing product variability by virtual products, *International Journal of Production Research*, 35(8): 2093-2107.

Otto, K., Gonzalez-Zugasti, J., and Dahmus, J., 2000, Modular product architecture, ASME Design Engineering Technical Conferences, DETC2000/DTM-4565, Baltimore, MD.

Otto, K., Tang, V., and Seering, W., 2003, Establishing quantitative economic value for features and functionality of new products and new services, Chapter N, MIT PDMA Toolbook II, http://hdl.handle.net/1721.1/3821 .

Park, J., and Simpson, T.W., 2005, Development of a production cost estimation framework to support product family design, *International Journal of Production Research*, 43(4): 731-772.

Pekelman, D., and Sen, S., 1979, Measurement and estimation of conjoint utility functions, *Journal of Consumer Research*, 5(4): 263-271.

Piller, F.T., Moeslein, K., and Stotko, C.M., 2004, Does mass customization pay? An economic approach to evaluate customer integration, *Production Planning & Control*, 15(4): 435-444.

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

Pine, B.J., Victor, B., and Boynton, A.C., 1993, Making mass customization work, *Harvard Business Review*, 71(5): 108-121.

Porter, M.F., 1980, An algorithm for suffix stripping, *Program*, 14(3): 130-137.

Prasad, B., 1996, *Concurrent Engineering Fundamentals*, Prentice Hall PTR, NJ.

Prasad, B., 1998, Designing products for variety and how to manage complexity, *Journal of Product & Brand Management,* 7(3): 208-222.

Prebil, I., Zupan, S., and Lucic, P., 1995, Adaptive and variant design of rotational connections, *Engineering with Computers*, 11(2): 83-93.

Prudhomme, G., Zwolinski, P., and Brissaud, D., 2003, Integrating into the design process the needs of those involved in the product life-cycle, *Journal of Engineering Design*, 14(3): 333-353.

Pullman, M.E., Moore, W.L., and Wardell, D.G., 2002, A comparison of quality function deployment and conjoint analysis in new product design, *Journal of Product Innovation Management*, 19(5): 354-364.

Pyle, D., 1999, *Data Preparation for Data Mining*, Morgan Kaufmann, San Francisco, CA.

Qiu, S.L., Fok, S.C., Chen, C.H., and Xu, S., 2002, Conceptual design using evolutionary strategy, *Advanced Manufacturing Technology*, 20(9): 683-691.

Raman, N., and Chhajed, D., 1995, Simultaneous determination of product attributes and prices and production processes in product-line design, *Journal of Operations Management*, 12(3-4): 187-204.

Ramdas, K., Fisher, M., and Ulrich, K., 2003, Managing variety for assembled products: Modeling component systems sharing, *Manufacturing & Service Operations Management*, 5(2): 142-156.

Ramdas, K., and Sawhney, M.S., 2001, A cross-functional approach to evaluating multiple line extensions for assembled products, *Management Science*, 47(1): 22-36.

Roberson, J., 1999, Designing effective portfolio variety using customer need discrimination thresholds, M.S. Thesis, Massachusetts Institute of Technology, Cambridge, MA.

Robertson, D., and Ulrich, K., 1998, Planning product platforms, *Sloan Management Review*, 39(4): 19-31.

Robinson, W.T., 1988, Marketing mix reactions to entry, *Marketing Science*, 7(4): 368-385.

Rothwell, R., and Gardiner, P., 1990, Robustness and product design families, In *Design Management: A Handbook of Issues and Methods*, Oakley, M. (Ed.), pp. 279-292, Basil Blackwell, Cambridge, MA.

Rugg, G., and McGeorge, P., 1995, Laddering, *Expert System*, 12(4):279-291.

Saari, D.G., 2000, Mathematical structure of voting paradoxes: I. Pairwise votes, *Economic Theory*, 15(1): 1-53.

Saaty, T., 1980, *The Analytic Hierarchy Process*, McGraw-Hill, New York.

Sabin, D., and Weigel, R., 1998, Product configuration frameworks - A survey, *IEEE Intelligent Systems & Their Applications*, 13(4): 42-49.

Safizadeh, M.H., Ritzman, L.P., and Mallick, D., 2000, Revisiting alternative theoretical paradigms in manufacturing strategy, *Production and Operations Management*, 9(2): 111-127.

Salhieh, S.M., and Kamrani, A.K., 1999, Macro level product development using design for modularity, *Robotics and Computer Integrated-Manufacturing*, 15(11): 319-329.

Sanchez, R., 1994, Towards a science of strategic product design: System design, component modularity, and product leveraging strategies, The 2nd International Product Development Management Conference on New Approaches to Development and Engineering, Gothenburg, Sweden.

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

Sarwar, B., 2001, Sparsity, scalability, and distribution in recommender systems, PhD thesis, University of Minnesota, MN.

Sawhney, M.S., 1998, Leveraged high-variety strategies: From portfolio thinking to platform thinking, *Journal of the Academy of Marketing Science*, 26(1): 54-61.

Seepersad, C.C., Mistree, F., and Allen, J.K., 2002, A quantitative approach for designing multiple product platforms for an evolving portfolio of products, ASME Design Engineering Technical Conferences, DETC2002/DAC-34096, Montreal, Quebec, Canada.

Sharman, D.M., and Yassine, A.A., 2004, Characterizing complex product architectures, *Systems Engineering*, 7(1): 35-60.

Shaw, M.L.G., and Gaines, B.R., 1996, Requirements acquisition, *Software Engineering Journal*, 11(3): 149-165.

Shocker, A.D., and Srinivasan, V., 1979, Multiattribute approaches for product concept evaluation and generation: A critical review, *Journal of Marketing Research*, 16(2): 159-180.

Shoji, S., Graham, A., and Walden, D., 1993, *A New American TQM*, Productivity Press, Portland, OR.

Siddique, Z., 2001, Estimating reduction in development time for implementing a product platform approach, ASME Design Engineering Technical Conferences, DETC2001/CIE-21238, Pittsburgh, PA.

Siddique, Z., 2005, Assembly process selection to minimize existing assembly system modification cost during new product family member design, ASME Design Engineering Technical Conferences, DETC2005-85016, Long Beach, CA.

Siddique, Z., and Repphun, B., 2001, Estimating cost savings when implementing a product platform approach, *Concurrent Engineering: Research and Application*, 9(4): 285-294.

Siddique, Z., and Rosen, D.W., 1999, Product platform design: a graph grammar approach, ASME Design Engineering Technical Conferences, DETC99/DTM-8762, Las Vegas, NV.

Siddique, Z., and Rosen, D.W., 2001, On discrete design spaces for the configuration design of product families, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 15(2): 91-108.

Siddique, Z., Rosen, D.W., and Wang, N., 1998, On the applicability of product variety design concepts to automotive platform commonality, ASME Design Engineering Technical Conferences, 98-DETC/DTM-5661, Atlanta, GA.

Silveira, G.D., Borenstein, D., and Fogliatto, F.S., 2001, Mass customization: Literature review and research directions, *International Journal of Production Economics*, 72: 1-13.

Simpson, T.W., 2004, Product platform design and customization: Status and promise, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *Special Issue on Platform Product Development for Mass Customization*, 18(1): 3-20.

Simpson, T.W., and D'Souza, B., 2004, Assessing variable levels of platform commonality within a product family using a multiobjective genetic algorithm, *Concurrent Engineering: Research and Applications,* 12(2): 119-130.

Simpson, T.W., Nanda, J., Halbe, S., Umapathy, K., and Hodge, B., 2003, Development of a framework for web-based product platform customization, *ASME Journal of Computing and Information Science in Engineering,* 3(2): 119-129.

Simpson, T.W., Siddique, Z., and Jiao, J., 2005, *Product Platform and Product Family Design: Methods and Applications*, Springer, New York.

Sosa, M.E., Eppinger, S.D., and Rowles, C.M., 2003, The misalignment of product architecture and organizational structure in complex product development, INSEAD Working Paper, 2003/68/TM.

Steiner, W.J., and Hruschka, H., 2002, A probabilistic one-step approach to the optimal product line design problem using conjoint and cost data, *Review of Marketing Science*, Working Papers, 1(4): Working Paper 4, http://www.bepress.com/roms/vol1/iss4/paper4.

Stone, R.B., Wood, K.L., and Crawford, R.H., 2000, A heuristic method for identifying modules for product architectures, *Design Studies*, 21(1): 5-31.

Sudharshan, D., May, J.H., and Shocker, A.D., 1987, A simulation comparison of methods for new product location, *Marketing Science*, 6(2): 182-201.

Suh, N.P., 2001, *Axiomatic Design - Advances and Applications*, Oxford University Press, New York.

Sundgren, N., 1999, Introducing interface management in product family development, *Journal of Production Innovation Management*, 16(1): 40-51.

Tarasewich, P., and Nair, S.K., 2001, Designer-moderated product design, *IEEE Transactions on Engineering Management*, 48(2): 175-188.

Thevenot, H.J., and Simpson, T.W., 2005, Commonality indices for assessing product families, In *Product Platform and Product Family Design: Methods and Applications*, Simpson, T.W., Siddique, Z., Jiao, J. (Eds.), pp. 107-129, Springer, New York.

Tiihonen, J., Lehtonen, T., Soininen, T., Pulkkinen, A., Sulonen, R., and Riitahuhta, A., 1998, Modeling configurable product families, The 4th WDK Workshop on Product Structuring, Delft University of Technology, Delft, The Netherlands.

Train, K.E., 2003, *Discrete Choice Methods with Simulation*, Cambridge University Press, UK.

Tseng, M.M., and Jiao, J., 1996, Design for mass customization, *CIRP Annals*, 45(1): 153-156.

Tseng, M.M., and Jiao, J., 1998, Computer-aided requirement management for product definition: A methodology and implementation, *Concurrent Engineering: Research and Application*, 6(2): 145-160.

Tseng, M.M., and Piller, F.T., 2003, The customer centric enterprise, in *The Customer Centric Enterprise: Advances in Mass Customization and Personalization*, Tseng, M., and Piller, F. (Eds.), pp. 3-16, Springer, NY.

Tsubone, H., Matsuura, H., and Satoh, S., 1994, Component part commonality and process flexibility effects on manufacturing performance, *International Journal of Production Research*, 32(10): 2479-2493.

Turksen, I.B., and Willson, I.A., 1992, Customer preferences models: Fuzzy theory approach, Proceedings of the SPIE - International Society for Optical Engineering, pp. 203-211, Boston, MA.

Ulrich, K., 1995, The role of product architecture in the manufacturing firm, *Research Policy*, 24(3): 419-440.

Ulrich, K., and Eppinger, S.D., 1995, *Product Design and Development*, McGraw-Hill, New York.

Ulrich, K., and Tung, K., 1991, Fundaments of product modularity, *Issues in Mechanical Design International,* Sharon, A. (Ed.), New York, ASME, DE-39, pp. 73-79.

Urban, G.L., and Hauser, J.R., 1993, *Design and Marketing of New Products*, Prentice-Hall, NJ.

Van Veen, E.A., 1992, *Modeling Product Structures by Generic Bills-of-Materials*, Elsevier, New York.

Van Wie, M.J., Rajan, P., Campbell, M.I., Stone, R.B., and Wood, K.L., 2003, Representing product architecture, ASME Design Engineering Technical Conferences, DETC2003/DTM-48668, Chicago, IL.

Van Wie, M., Stone, R.B., Thevenot, H., and Simpson, T., 2006, Examination of platform and differentiating elements in product design, *Journal of Intelligent Manufacturing*, *Special Issue on Product Family Design and Development,* in press.

Veryzer, R.W. Ji., 1993, Aesthetic response and the influence of design principles on product performance, In *Advances in Customer Research*, McAllister, L., Rothschele, M. (Eds.), pp. 224-231, Provo, UT.

Wacker, J.G., and Trelevan, M., 1986, Component part standardization: An analysis of commonality sources and indices, *Journal of Operations Management*, 6(2): 219-244.

Warren, A.A., 1983, Optimal control of the product portfolio, Ph.D. Thesis, The University of Texas at Austin, Austin, TX.

Wassenaar, H.J., and Chen, W., 2001, An approach to decision-based design, Proceedings ASME 2001 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, DETC2001/DTM-21683, Pittsburgh, PA.

Watson, A., Viney, H., and Schomaker, P., 2002, Consumer attitudes to utility products: A consumer behaviour perspective, *Marketing Intelligence & Planning*, 20(7): 394-404.

Wedel, M., and Kamakura, W.A., 1998, *Market Segmentation: Conceptual and Methodological Foundations*, Kluwer, Boston, MA.

Weir, O., 2000, Analysis of customer-driven and systemic variation in the airplane assembly process, M.S. Thesis, Massachusetts Institute of Technology, Cambridge, MA.

Whitney D.E., 2003, Physical limits to modularity, Working paper, ESD-WP-2003-01.03-ESD, Massachusetts Institute of Technology, Cambridge, MA.

Wilhelm, B., 1997, Platform and modular concepts at Volkswagen – Their effects on the assembly process, In *Transforming Automobile Assembly*, Shimokawa, K., Jürgens, U., Fujimoto, T. (Eds.), Springer - Verlag, Berlin Heidelberg.

Wittink, D.R., and Cattin, P., 1989, Commercial use of conjoint analysis: An update, *Journal of Marketing*, 53(2): 91-96.

Wortmann, J.C., Muntslag, D.R., and Timmermans, P.J.M., 1997, *Customer Driven Manufacturing*, Chapman & Hall, London.

Yan, W., Chen, C.H., and Khoo, L.P., 2001, A radial basis function neural network multicultural factors evaluation engine for product concept development, *Expert System*, 8(5): 219-232.

Yan, W., Chen, C.H., and Khoo, L.P., 2002, An integrated approach to the elicitation of customer requirements for engineering design using picture sorts and fuzzy evaluation, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 16(2): 59-71.

Yang, L., and Gao, Y., 1996, *Fuzzy Mathematics: Theory and Applications*, HNSTU Press, Guangzhou, China.

Yano, C., and Dobson, G., 1998, Profit optimizing product line design, selection and pricing with manufacturing cost considerations, in *Product Variety Management: Research Advances*, Ho, T.-H., Tang, C.S. (Eds.), pp. 145-176, Kluwer Academic Publisher, Boston, MA.

Yong, S.K., Yum, B.-J., Song, J., and Su, M.K., 2005, Development of a recommender system based on navigational and behavioral patterns of customers in e-commerce sites, *Expert Systems with Applications*, 28(2): 381-393.

Yoon, H.C., and Jae, K.K., 2004, Application of Web usage mining and product taxonomy to collaborative recommendations in e-commerce, *Expert Systems with Applications*, 26(2): 233-246.

Yu, J.S., Gonzalez-Zugasti, J.P., and Otto, K.N., 1999, Product architecture definition based upon customer demands, *ASME Journal of Mechanical Design*, 121(3): 329-335.

Yu, T.-L., Yassine, A.A., and Goldberg, D.E., 2003, A genetic algorithm for developing modular product architectures, ASME Design Engineering Technical Conferences, Chicago, IL.

Yuan, S.-T., and Cheng, C., 2004, Ontology-based personalized couple clustering for heterogeneous product recommendation in mobile marketing, *Expert Systems with Applications*, 26(4): 461-476.

Zaltman, G., 2003, *How Customers Think: Essential Insights into the Mind of the Market*, Harvard Business School Press, Boston, MA.

Zamirowski, E.J., and Otto, K.N., 1999, Identifying product portfolio architecture modularity using function and variety heuristics, ASME Design Engineering Technical Conferences, DETC99/DTM-876, Las Vegas, NV.

Zeithaml, V.A., 1988, Consumer perceptions of price, quality, and value: A means-end model and synthesis of evidence, *Journal of Marketing*, 52(1): 2-22.

Zeithaml, V.A., and Bitner, M.J., 2001, *Services Marketing: Integrating Customer Focus across the Firm*, China Machine Press, Beijing, China.

Zha, X.F., Sriram, R.D., and Lu, W.F., 2004, Evaluation and selection in product design for mass customization: A knowledge decision support approach, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 18(1): 87-109.

Zhang, J., Wang, Q., Wan, L., and Zhong, Y., 2005, Configuration-oriented product modeling and knowledge management for made-to-order manufacturing enterprises, *International Journal of Advanced Manufacturing Technology,* 25(1-2): 41-52.

Zimmermann, H.J., 1985, *Fuzzy Set Theory and Its Applications*, Kluwer-Nijhoff Publishing, Boston, MA.