GAME THEORETIC OPTIMIZATION FOR PRODUCT LINE EVOLUTION

A Master Thesis
Presented to
The Academic Faculty

By

Ruoyu Song

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

Georgia Institute of Technology

December, 2015

Copyright © 2015 by Ruoyu Song

Game Theoretic Optimization for Product Line Evolution

Approved by:

Dr. Roger J. Jiao, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Cassandra Telenko
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Manpreet S. Hora
College of Business
Georgia Institute of Technology

Date Approved: December 2, 2015

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor Dr. Jiao for the continuous guidance and support of my master study and research, for his patience and motivation. He offered invaluable insights into the topic, scope and in dealing with research problems. His guidance helped me in all the time of research and writing of this thesis. I would also like to thank Dr. Telenko and Dr. Hora. As my thesis reading committee members, they provided insightful comments and suggestions for my thesis.

Finally, I would like to thank my parents and my elder sister. They were always supporting me and encouraging me with their best wishes.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

a_k= product attribute;

A= set of product attributes;

A^{Active} = set of updated attributes

a_{il}^*= attribute level;

A_i^*= set of attribute levels for a attribute;

p_j = product profile;

P = set of products;

Ω^{Exist} = existing product line;

Ω^{New} = new product line;

C_j = engineering cost of a product;

s_k = market segment;

S =set of market segments;

Q_k = size of a market segment;

U_{kj} = customer-perceived utility of a product;

D_{kj} = probability of a customer choosing a product;

u_{kil} = part-worth utility of segment s_k for the *l*th level of attribute a_i;

w_{ji} = utility weights among attributes;

π_j= constant of composite utility for a product;

ε_{kj} = error term for a segment-product pair;

μ = scaling parameter of conditional multinomial logit choice rule;

PCI_j= process capability index corresponding to a product;

LSL^T = lower specification limit of cycle time estimation;

μ_j^T= mean value of the estimated cycle time corresponding to a product;

σ_j^T= standard deviation of the estimated cycle time corresponding to a product;

β = constant indicating the average dollar cost per variation of process capabilities;

ζ_{ji}/ω_j = regression coefficients;

μ_{il}^{t}= mean value of the part-worth standard time for attribute level a_{kl}^{*};

σ_{il}^{t}= standard deviation of the part-worth standard time for attribute level a_{kl}^{*};

x_{jil} = integer variable indicating the choice of an attribute level in a product;

y_j = binary variable indicating the decision of offering of a product.

X = set of integer variables indicating each choice of an attribute level for potential products

Y = set of binary variables indicating each decision of offering a product for product line

SUMMARY

Product line planning aims at optimal planning of product variety. In addition, the traditional product line planning problem develops new product lines based on product attributes without considering existing product lines. However, in reality, almost all new product lines evolve from existing product lines, which leads to the product line evolution problem. Product line evolution involves trade-offs between the marketing perspective and engineering perspective. The marketing concern focuses on maximizing utility for customers; the engineering concern focuses on minimizing engineering cost. Utility represents satisfaction experienced by the customers of a product. Engineering cost is the total cost involved in the process of the development of a product line. These two goals are in conflict since the high utility requires high-end product attributes which could increase the engineering cost and vice versa. Rather than aggregating both problems as one single level optimization problem, the marketing and engineering concerns entail a non-collaborative game per se. This research investigates a game-theoretic approach to the product line evolution problem. A leader-follower joint optimization model is developed to leverage conflicting goals of marketing and engineering concerns within a coherent framework of game theoretic optimization. To solve the joint optimization model efficiently, a bi-level nested genetic algorithm is developed. A case study of smart watch product line evolution is reported to illustrate the feasibility and potential of the proposed approach.

CHAPTER 1: INTRODUCTION

This chapter introduces the motivation of this research in section 1.1, defines the research objectives and scope in section 1.2, and then outlines the organization of this thesis in section 1.3.

1.1 Motivation

1.1.1 Mass customization

By taking the benefit of mass production efficiency, mass customization introduces product proliferation, which satisfies increasing diversification of customer needs and improves the sales (Pine 1993). However, the benefits do not keep increasing as variety increasing. Variety can cause exponential growth of complexity and diminish the efficiency of manufacturing processes, thus the engineering costs would increase; it can also jeopardize the efficiency of the manufacturing process (Wortmann et al., 1997). In addition, mass customization may lead to mass confusion (Huffman and Kahn, 1998). A wide variety of products provides customers too many choices, which is more than the customer needs. Therefore, companies should offer proper product varieties to customers (Pine et al., 1993). For example, Nissan reportedly offered 87 different types of steering wheels; 20% of Toyota's product variety accounted for 80% of its sales. Moreover, different markets may have different customer needs. Therefore, companies should offer different product varieties to different target markets.

The company should make the decision on product varieties at the early stage of the entire product design process because this is a key decision. Once this decision is fixed, large amount of costs are committed for the remainder of the design process. Therefore, the quality of the decision made on product varieties is very important.

A product is customized by choosing different attributes. A product attribute is an element for a product. It is also called option or feature for product. Therefore, a product is composed of various attributes. For example, the CPU type, storage size and battery life are all attributes of a laptop.

Product line is a set of products with similar functions but having some different characteristics offered by the same company. Traditional product line planning problem develops new product line based on attributes without considering existing product line. However, in reality, almost all new product lines evolve from existing product line, which leads to product line evolution problem. For a product line evolution, the existing product line should serve as parameters when developing new product lines.

1.1.3 Marketing-engineering trade-off

There are two perspectives for product line evolution, marketing perspective and engineering perspective. Traditionally, product line planning focuses on marketing perspective. The goal for product line planning is to maximize the profit, sale or customer perceived value (utility). Therefore, it is necessary to measure the customer preference. Conjoint analysis is one of the most popular preference-based techniques for identifying and evaluating new product concepts (Green and Krieger, 1985; 1996.). From the engineering perspective, the main goal is to minimize the engineering cost. It is very difficult to measure the exact engineering costs at the early stage of product lifecycle.

Previously, marketing and engineering perspectives are considered as a collaborative problem, which means they can achieve global optimization results to satisfy both requirements. However, maximizing customer perceived value and minimizing engineering costs are non-collaborative, which means the goals of the marketing and engineering perspectives are in conflict. For example, assume a high-end computer and a low-end computer are at the same price. The computer with high-end configurations such as an i7-CPU, 16GB memory and 1TB SSD could provide higher customer perceived value than the low-end computer with i3-CPU, 4GB memory and 512GB HDD. However, the engineering cost for a low-end computer is much less than the high-end one. From the marketing perspective, high-end computers are preferred since customers have higher preferences to purchase them due to high utilities. From the engineering perspective, low-end computers are preferred since engineering cost including the design and manufacturing cost is less. Therefore, the optimization should trade off the marketing perspective and engineering perspective.

1.2 Research objectives and scope

The objective of this research is to develop a game theoretic optimization for product line evolution to assist company to provide right choices of products. This research introduces the evolution process into the product line planning problem, which most previous literature does not cover. In addition, the product line evolution problem is treated as a non-collaborative game in this research, since the goals of marketing and engineering perspectives are in conflict. There are two stages for product line evolution, which are the product line generation stage and the optimization stage (Li and Azarm, 2002).This research focuses on the optimization stage of the product line evolution, which means that the attributes need to be optimized and their information are all provided. The optimization for the product line evolution is a combinatorial optimization problem. A bi-level nested genetic algorithm (BNGA) is developed in this thesis to solve this problem. A case study of smart watch is reported to illustrate the feasibility and the potential of the proposed approach. This research targets the competitive markets where the producers need to improve their profits to survive. The profit in this research is not measured directly. The profit is considered maximized when the utility is maximized while the engineering cost is minimized. To evaluate the quality of the results, the utility/cost ratio for the product line should be computed. The larger the ratio is, the better the results are. This research does not consider the competitors' decisions. All decisions are made based on the company's existing product line and new added attributes. The customer perceived values are calculated based on the part-worth utility, customer preference and the demand quantity for specific market segments. The engineering costs are calculated based on standard time estimation (Tseng and Jiao, 1996), and the demand for each product.

1.3 Organization of the thesis

There are in total 7 chapters for this thesis. Chapter 1 is a general introduction to this research which illustrates the motivation of product line evolution. Chapter 2 is the literature review on product line planning and evolution from both marketing and engineering perspective. The literature review also covers the design optimization and then the game theory for product line planning. Chapter 3 covers the problem formulation for game theoretic decisions on product line evolution. This chapter develops a product line evolution model with marketing and engineering interaction. Since the product line evolution is a non-collaborative game, it is formulated as a leader-follower joint optimization. In chapter 4, the methodology to deal with leader-follower joint optimization of marketing and engineering is developed. For this methodology, bi-level nested genetic algorithm is adopted as the solution which is described in chapter 5. Chapter 6 uses the smart watch as a use case to validate the model and the solution strategy for product line evolution. Chapter 7 presents the conclusion for the research, and also discusses the limitations and future work for this research.

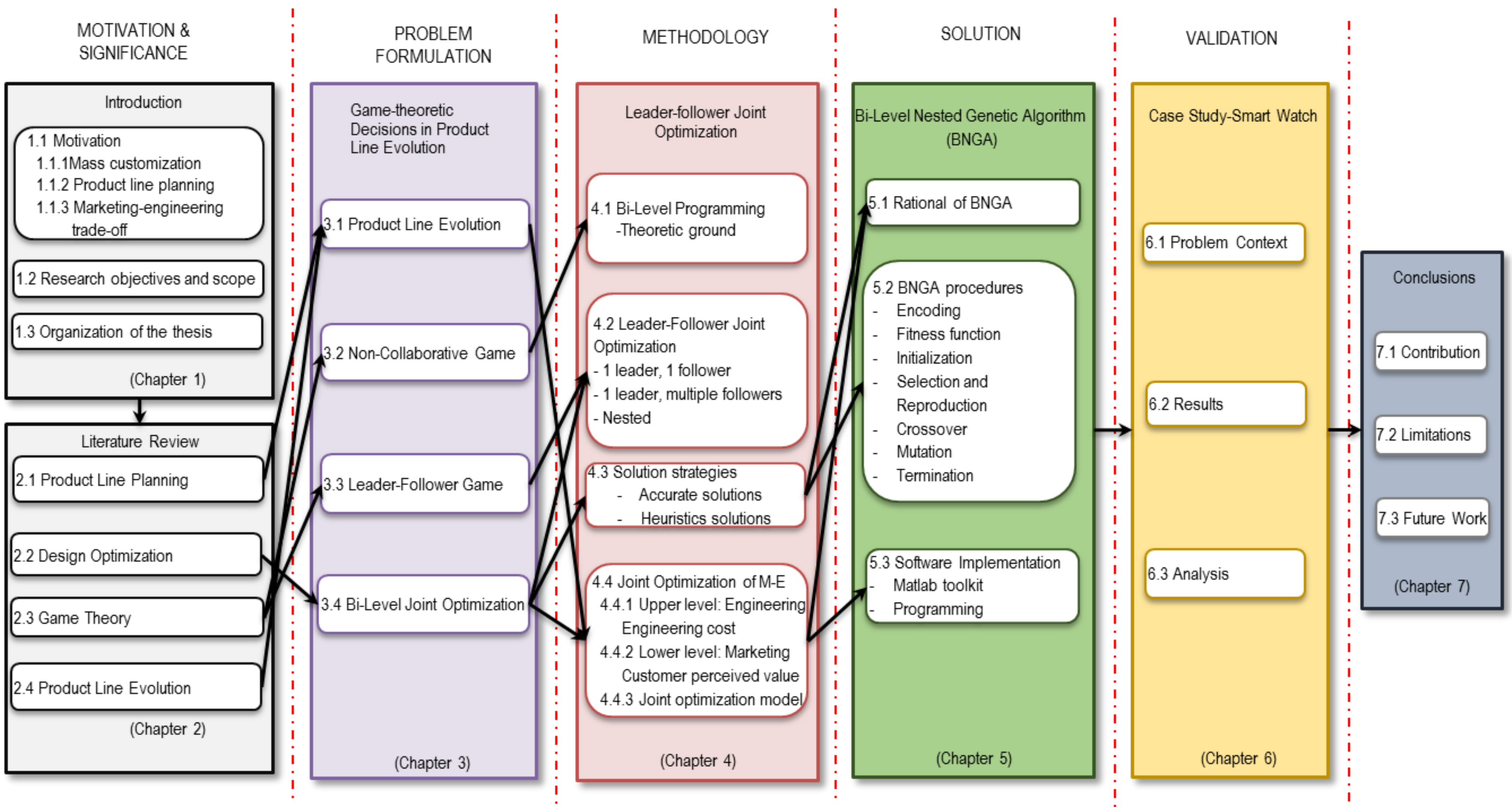

Figure 1-1. Research Roadmap

CHAPTER 2: LITERATURE REVIEW

This chapter systematically reviews the literature from four perspectives. Section 2.1 reviews the product line planning. Section 2.2 reviews the design optimization for product line planning problem. Section 2.3 summarizes the game theory used for design optimization. And Section 2.4 reviews current theory and methodology for product line evolution process.

2.1 Product line planning

For the product line planning, most literature maximizes the surplus in order to get the optimal combination of different products. The surplus is the margin between the customer perceived value and the price of the products (Kaul and Rao, 1995; Kohli and Sukumar, 1990). There are also other goals used to select products among potential products such as maximization of profit (Monroe et al., 1976), market share (Kohli and Krishnamurti, 1987), a seller's welfare (McBride and Zufryden, 1988), share of choices (Balakrishnan and Jacob, 1996), and net present value (Li and Azarm, 2002).

QFD and conjoint analysis have been combined to compare the most preferred features with profit maximizing features to optimize the sales or profits of product line (Pullmana et al., 2002). Product line commonality measure for product line can capture the level of component commonality in a product family in order to minimize non-value added variations across models without limiting customer choices (Kota et al., 2002).

For the product line, most papers are in marketing literature, which focus on the marketing perspective to maximize profit. However, very few of them model the costs of engineering design (Yano and Dobson, 1998). Green and Krieger (1985) did not incorporate prices or costs in their product line models. Then Dobson and Kalish (1988, 1993) extended to include per-product fixed cost. Product line design models include more complex cost structures recently. Raman and Chhajed (1995) and Kim and Chhajed (2001) have noticed that the product line planning must choose the manufacturing process for the products in addition to choosing which products to offer. Dobson and Yano (1994) studied complex interaction by considering per-product fixed costs, resources shared by multiple products, and technology choices for each. Radas and Sayney (2001) also analyze the fixed cost of a component shared by two products. Chidambaram and Agogino (1999) have formulated the product line analysis as an optimization problem which is consistent with the manufacturer's goal to minimize costs in the redesign of existing standard components, and meet the customer specifications and design constraints at the same time.

In addition, product line planning involves two stages: generation of a set of feasible product alternatives, and construction of a product line by selecting products from the reference set (Li and Azarm, 2002). There are two categories of existing approaches to product line planning which are one-step approaches and two-step approaches (Steiner and Hruschka, 2002). One-step approaches use part-worth preference and cost-return functions to construct the product line directly. Two-step approaches reduce the entire set of feasible product profiles to a smaller set first, then select products from this reduced set to construct a product line. Kohli and Sukumar (1990) and Nair et al. (1995) used one-step approaches. However, most literature adopts the two-step approaches and focus on maximizing the profit in the second step (McBride and Zufryden, 1988; Dobson and Kalish, 1993; Chen and Hausman, 2000). Generally speaking, the one-step approach is better than two-step approach since the intermediate step for the two-step approach of enumerating utilities and profits of a large quantity of reference set product profiles can be eliminated by using one-step approach (Steiner and Hruschka, 2002). The two-step approach works better only when the reference set contains small number of product profiles. Hence, most literature only involves small number of attributes to describe the product (Yano and Dobson, 1998).

2.2 Design optimization

Traditional design optimization emphasizes more on the designer's perspective (Tarasewich and Nair, 2001). The primary concern of product design optimization is to measure customer preference in terms of expected utilities (Krishnan and Ulrich, 2001). For preference-based product design, conjoint analysis (Green and Krieger, 1985) has been proven to be an effective method to estimate part-worth utilities of individual attribute levels with individual product attributes. The market shares of potential products can be simulated by collecting scaled preference evaluations from respondents with regard to a subset of a subset of product profiles with multiple attributes which is constructed according to fractional factorial design. Then, the part-worth preference functions could be estimated using regression analysis. The part-worth utilities of attribute level can also be measured using choice-based conjoint analysis, and then establishes a direct connection between preference and choice (Kuhfeld, 2004). Generally, conjoint analysis uses discrete attribute, and thus the product design optimization using conjoint-based searching is always a combinatorial optimization problem (Kual and Rao, 1995; Kohli and Sukumar, 1990; Nair et al., 1995).

Multi-attribute utility analysis is widely used to predict total utilities for feasible product profile composed of underlying attribute level part-worth utilities (Keeney and Raiffa, 1976). Multi-attribute utility analysis assumes that the utility of each attribute are mutually independent (Wassenaar and Chen, 2001). However, for the product line, this may not be true because the customer perceived value (utility) of an individual attribute may vary due to the availability of other offerings.

Furthermore, multi-attribute weighting and normalization must be considered when combining individual attribute utility functions into one multi-attribute utility function. The weight for each attribute should be determined based on the rank order of alternatives. But a selected alternative may result from the underlying voting method rather than the quality of the alternative itself (Saari, 2000). Normalization is often used to assist the comparison of design alternatives when the attributes have different metrics or dimensions. The normalized value depends on the relative position of the attribute level within the range of values, and thus, there is not a rigorous method to determine the normalizing range (Wassenaar and Chen, 2001). The weighted sum method which assigns different weights to attributes is often adopted for the relative importance of multiple attributes. However, the weights assignment is subjective and often results in bias when an attribute is correlated to a product's success (Arrow and Raynaud, 1986). In addition, the weight sum method assumes a linear tradeoff, which is only true for limited variation of attribute levels (Wassenaar and Chen, 2001). Therefore, this method is not suitable for product line planning since the number of attributes and levels could be large. Wassenaar and Chen (2001) have addressed the necessity to use a single criterion approach to decision-based design, which should reflect various issues regarding customers, design and manufacturing.

2.3 Game theory

The product line planning problem needs to address two concerns which are marketing and engineering. Previously, the marketing perspective and engineering perspective are combined into an all-in-one problem, and then to find the global optimal results (Jiao and Zhang, 2005). However, these two perspectives are non-collaborative. Therefore, product line planning should be a bi-level optimization problem. It is originated from game theory with a hierarchical optimization problem (Stackelberg, 1952). In the bi-level optimization problem, the lower-level optimization problem which is the follower is nested within the upper-level optimization problem which is the leader (Colson et al., 2007). The leader and the follower compete against each other; the leader makes the decision first and then the follower reacts based on the leader's decision to make his/her optimal decision as feedback, and then the leader adjusts the decision accordingly (Du et al., 2014). The process terminates when both actors obtain satisfactory solutions. However, the solution to the bi-level programming is hard to obtain, mainly because of non-convexity (Calvete et al., 2008). Traditional solutions usually replace the lower-level problem with its Karush-Kuhn-Tucker (KKT) conditions which are the conditions when it is convex and continuous differentiable, and then applying gradient methods (Calvete et al., 2008). However, this type of approaches is not efficient for large problems, and may lead to a paradox that the follower's decision dominates the leader's (Lai, 1996). Recently, evolutionary algorithms are adopted to deal with complex optimization problems such as genetic algorithms. Evolutionary algorithms usually have low risks of ending up in a local optimum (Brands and van Berkum, 2014). Calvete et al. (2008) combine classical enumeration techniques with genetic algorithm to achieve near-optimal solutions in reasonable computational times. Ji et al. (2013) adopts a constrained genetic algorithm to solve a leader-follower joint optimization problem of technical system modularity and material reuse modularity.

2.4 Product line evolution

Traditionally, product line planning is treated as a static optimization problem. In practice, companies usually update the product lines by introducing new products and retiring old ones gradually, which is a process that mimics the natural evolution (Tellis and Merle Crawford, 1981, Sorenson 2000). The evolution concept has been recognized as a general design methodology for a wide range of applications (Hingston et al., 2008). Its application for product design has been emphasized on individual product (Otto and Wood, 1998; Maher and Tang, 2003), but few on product line planning. Ramdas and Sawhney (2001) utilized incremental revenue and life-cycle costs to evaluate multiple product line extensions. Bryan et al. (2007) proposed a co-evolution model with the goal of maximizing incremental profit for joint design of product lines and assembly system configurations.

2.5 Chapter summary

Previously, product line planning is based on individual attributes. The new product line does not consider the existing product. However, in reality, almost all new product lines are developed from existing product lines. Therefore, the input for the new product line should be the existing product line.

CHAPTER 3: GAME THEORETIC DECISIONS FOR PRODUCT LINE EVOLUTION

In this chapter, the problem formulation for product line evolution is represented in section 3.1, which illustrates the product line evolution process and clarifies the design factors. Section 3.2 shows that the product line evolution problem is a non-collaborative game. More specifically, it is a leader-follower game which is described in section 3.3. Section 3.4 formulates the problem as a bi-level joint optimization problem and defines the design variables for this problem.

3.1 Product line evolution

For the existing product line, each attribute is a_i, and a set of product attributes, $A = \{a_i | i = 1, \dots, I\}$ are identified, where i is the index of each attribute, and I is the total number of available attributes. The attributes may have several levels as $A_i^* = \{a_{il}^* | l = 1, \dots, L_i\}$, where l is the index of each attribute level, and L_i is the total number of attribute levels for each attribute. Each product is p_j, and the potential product profiles, $P^{Exist} = \left\{p_j^{Exist} \middle| j = 1, \dots, J\right\}$ are generated by selecting the attributes, where j is the index of each product, and J is the total number of potential products. A product line Ω^{Exist}, is a set consisting of several selected product profiles, i.e., $\Omega^{Exist} = \left\{p_j^{Exist} \middle| j = 1, \dots, J^*\right\} \subseteq P^{Exist}, \exists J^* \in \{1, \dots, J\}$, where J^* is the number of products for the product line. For the new product line, N new product attributes are added to the existing attributes. Therefore, the updated product attribute set is $A^{Active} = \{a_i | i = 1, \dots, I + N\}$. The new potential product profiles, $P^{New} = \left\{p_j^{New} \middle| j = 1, \dots, J\right\}$, are generated by selecting the updated attributes. The new product line, Ω^{New}, consists of new selected product profiles, i.e., $\Omega^{New} = \left\{p_j^{New} \middle| j = 1, \dots, J^*\right\} \subseteq P^{New}$. Figure 3-1 illustrates the product line evolution.

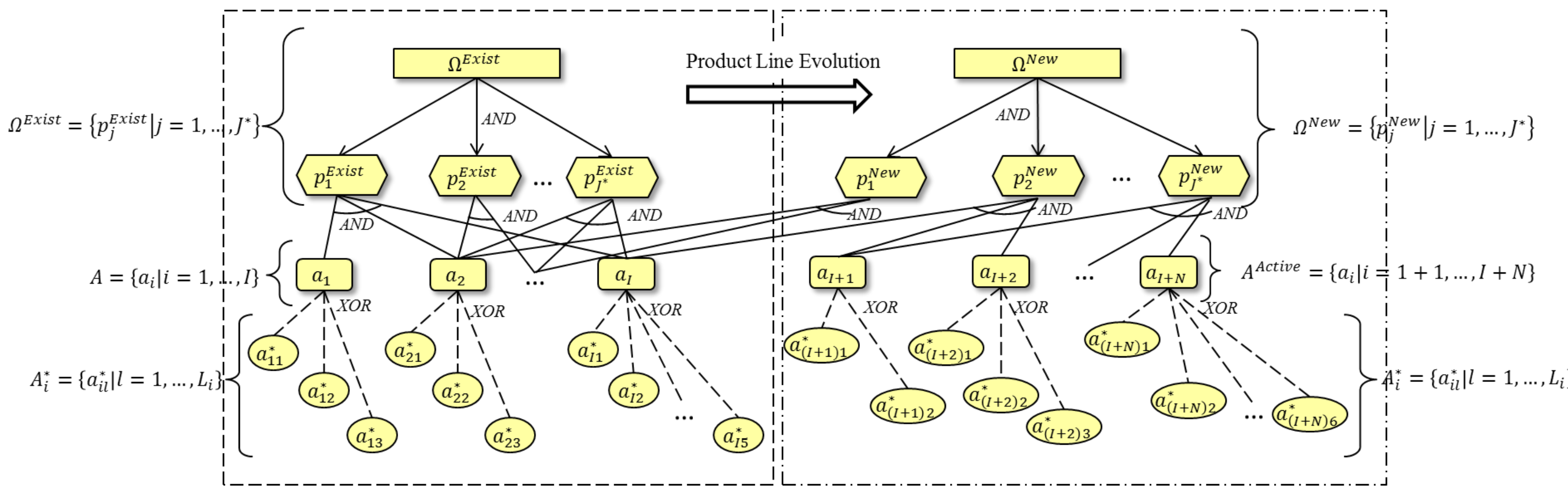

Figure 3-1. Evolution of Existing Product Line Ω^{Exist} to New

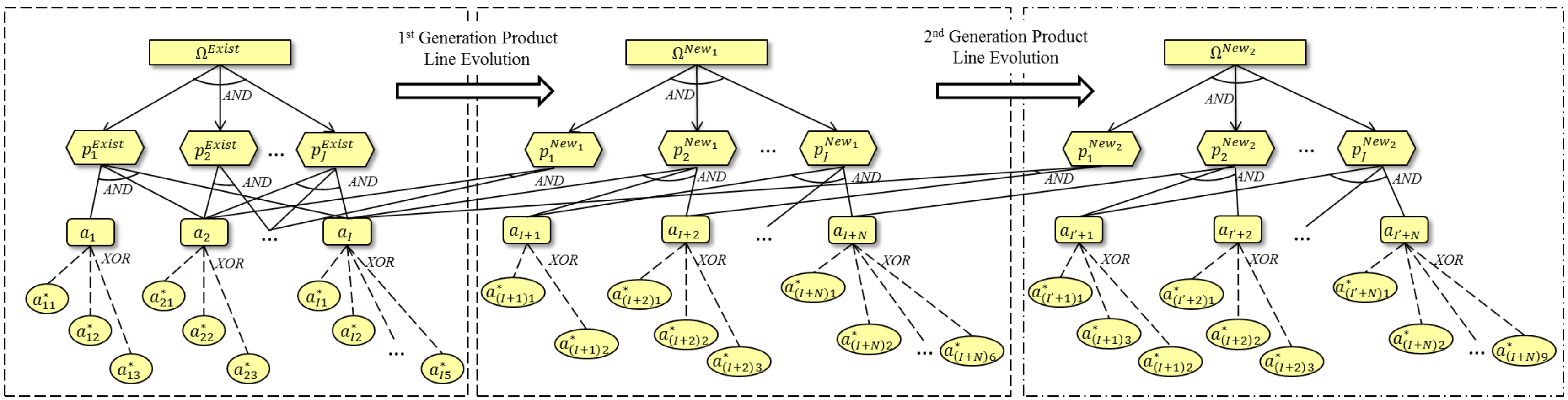

Figure 3-2. Evolution of 1st Generation Product Line to 2nd

The product line evolution is generic and easy to implement. As shown in the figure, the new set of attributes is added to the existing set of attributes to generate the updated attribute set A^{Active}. The new product line Ω^{New} consists of new product profiles that are combination of selected attributes. Compared to the existing product line Ω^{Exist}, the new product line may drop some of the existing attributes and add new attributes. Some products in Ω^{New}may be identical to the products in Ω^{Exist}. In addition, the product line evolution can serve future generations by continuously adding new product attributes to the updated feature set. Figure 3-2 shows the product line evolution for the second generation. Ω^{Exist} and $\Omega^{New_1}$are combined as the new Ω^{Exist}, new attributes are also given. $\Omega^{New_2}$is the product line for the second generation.

Figure 3-3 shows the evolution of product line for future generations. For the first generation product line, $\Omega^{Exist}$is given, Ω^{New_1}needs to be found. For the second generation, Ω^{Exist} and Ω^{New_1} are given, Ω^{New_2}needs to be found. The future generations follow this.

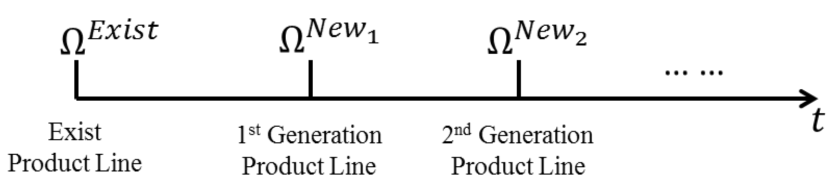

Figure 3-3. Product Line Evolution by Generations

From the marketing perspective, there are several market segments, $S = \{s_k | k = 1, \dots, K\}$. Each market segment contains customers with size Q_k. The customer preferences on different products are described by part-worth utilities, $\{U_{kj}\}_{KJ}$. The demand of the diverse products, $\{D_{kj}\}_{KJ}$, is represented by the customer choice probabilities. From the engineering perspective, each product has engineering cost, $\{C_j\}_J$, and price $\{\$_j\}_J$. The goal of the product line planning is to find the optimal combinations of attributes and their level of respective market segments to maximize the customer perceived value for marketing and minimize the engineering cost for engineering.

3.2 Non-collaborative game

The goal of the marketing perspective is to maximize the customer perceived value. In general, higher customer perceived value leads to higher engineering costs. However, the goal of the engineering perspective is to minimize the engineering costs. The decrease of the engineering costs generally diminishes the customer perceived value. Therefore, the goals of the marketing concern and engineering concern conflict with each other. Thus, they formulate a non-collaborative game. Traditionally, the product line planning problem is formulated to an all-in-one problem, which means the engineering and marketing perspectives are combined into one problem. Then a global optimal result can be found for this all-in-one problem. However, there is no global optimal result that can be generated for a non-collaborative game. The optimization process must leverage between the two concerns in order to generate an equilibrium result.

3.3 Leader-follower game

There are various types of games for non-collaborative game. For the marketing-engineering interaction, the non-collaborative game has two levels. Therefore, this is a bi-level competition. The bi-level competition is described by Stackelberg game that involves one leader and one follower. Therefore, leader-follower game is adopted in this research. The marketing perspective is treated as the leader (upper level). The engineering perspective is treated as the follower

(lower level). The leader-follower game assumes that both actors have certain decision power, and the leader has higher priority than the follower.

3.4 Bi-level joint optimization

The product line evolution involves a bi-level joint optimization problem. It consists of an upper level optimization problem and a lower level optimization problem. For the product line, maximizing the customer perceived value (upper level) and minimizing the engineering cost (lower level) should be joint together. Figure 3-4 shows that the optimization should leverage both marketing and engineering perspective. The upper level needs to find an optimal set of attributes and their levels to maximize the customer perceived value. Therefore, the leader has the design variable, X, which represents the choice of attributes and their levels. The lower level has the design variable Y to find an optimal set of products to minimize the engineering cost. The design variable Y is the decision of offering a product.

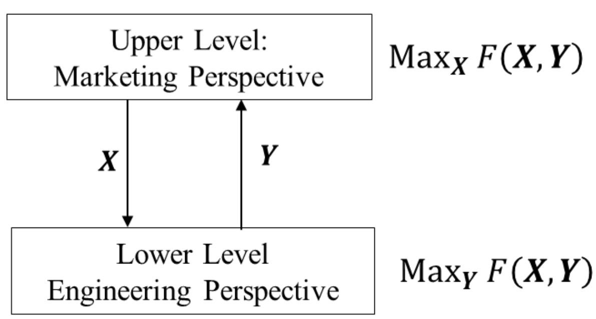

Figure 3-4. Bi-level Joint Optimization for Marketing-Engineering Interaction

3.5 Chapter Summary

This chapter formulates the product line evolution problem with description of all design factors including the attributes and their levels, the potential products and the existing and new product lines. The product line evolution problem is a non-collaborative game since maximizing the customer perceived value and minimizing the engineering cost has conflict goals. Furthermore, product line evolution is a leader-follower game. The marketing perspective is the leader, and the engineering perspective is the follower. Then the bi-level joint optimization model is formulated.

CHAPTER 4: LEADER-FOLLOWER JOINT OPTIMIZATION

This chapter illustrates the methodology to solve the problem formulated in chapter 3. Section 4.1 shows the theoretic ground to solve the bi-level optimization model. Section 4.2 describes different types of leader-follower joint optimization. Section 4.3 discusses solution strategies for the product line evolution. Section 4.4 illustrates detailed solution for the joint optimization of marketing and engineering interaction. This section describes upper level model, lower level model, and then the joint optimization model.

4.1 Bi-level programming

In general, the bi-level model formulation can be represented as follows (Bard, 1998; Colson et al., 2007):

$$\text{Max}_{\mathbf{X}}\ F(\mathbf{X}, \mathbf{Y}), \tag{4-1.1}$$

$$\text{s.t.}\quad G(\mathbf{X}, \mathbf{Y}) \leq 0, \mathbf{X} \in \mathbf{R}^s, \tag{4-1.2}$$

$$\mathbf{Y}\ \text{ is solved from} \tag{4-1.3}$$

$$\text{Max}_{\mathbf{Y}}\ f(\mathbf{X}, \mathbf{Y}), \tag{4-1.4}$$

$$\text{s.t.}\quad g(\mathbf{X}, \mathbf{Y}) \leq 0, \mathbf{Y} \in \mathbf{R}^t, \tag{4-1.5}$$

where $\mathbf{X}$ and F are the decision variables and objective functions for the upper level optimization problem; $\mathbf{Y}$ and f are the decision variables and objective functions for the lower level optimization problem. G and g are vector valued functions showing the constraints for the upper and lower level respectively. The bi-level optimization problem can be solved in three steps:

Step1: The leader makes a decision, $\boldsymbol{X}^*$, using his/her own strategy $F(\mathbf{X}, \mathbf{Y})$. And then informs the follower with this decision and a set of constraints;

Step 2: The follower makes a decision, $\boldsymbol{Y}^*$, according to his/her own strategy $f(\mathbf{X}, \mathbf{Y})$ and the leader's decision, $\boldsymbol{X}^*$. And then the follower gives the leader feedback of the decision $\boldsymbol{Y}^*$.

Step 3: The leader adjusts its decision to obtain a new decision $\boldsymbol{X}^{**}$ based on $F(\mathbf{X}, \mathbf{Y})$ and $\boldsymbol{Y}^*$.

These steps are iterated until both leader and follower achieve their satisfactory results (Ji et al., 2013). Equation (4-1) can be converted into a single-level parametric optimization problem (Colson et al., 2007):

$$\text{Max } F(\mathbf{X}, \mathbf{Y}'(\mathbf{X})), \tag{4-2.1}$$

$$\text{s.t.} \quad G(\mathbf{X}, \mathbf{Y}'(\mathbf{X})) \leq 0, \mathbf{X} \in \mathbf{R}^s. \tag{4-2.2}$$

It is converted based on the unique response function, $\mathbf{Y} = \mathbf{Y}'(\mathbf{X})$. Equation 3 is essentially a bi-level optimization problem. But it does not need to go through the three steps directly, and evolutionary algorithms are proposed to solve this problem efficiently (Li et al., 2014).

4.2 Leader-follower joint optimization

There are several types of leader-follower joint optimization.

1 leader-1 follower: there are only two actors in this model. The leader makes decision first, and then the follower reacts to this decision to make his/her optimal decision. The leader will change his/her decision based on the follower's decision. Therefore, the leader has higher priority than the follower.

1 leader-multiple followers: there are more than two actors in this model. The leader makes decision first, and then all followers (at least two) react to this decision to make their optimal decisions. The followers do not interchange information among themselves. The leader will then modify his/her decision based on all followers' decisions.

Nested: for the nested leader-follower joint optimization, it can be 1 leader-1 follower or 1 leader-multiple followers. For this type of optimization, after the leader modifies his/her decision, he will inform the follower again. Then the follower will also modify the decision. These steps are iterated until satisfactory results are achieved for both the leader and the follower.

For this research, the nested leader-follower joint optimization is used. There are 1 leader and 1 follower in the problem. In addition, the product line evolution should find the optimal results that both leader and follower are satisfied with results. 1 leader-1 follower joint optimization cannot fully meet the requirements. Therefore, nested leader-follower joint optimization is used.

4.3 Solution strategies

The formulation in equation 4-1 is an integer non-linear bi-level programming problem. This equation involves two design variables, which are the choice of attributes and the choice of products(X, Y). To solve the bi-level optimization problem, there are accurate solutions and heuristic solutions. One example of the accurate solutions is replacing the lower-level with its KKT conditions as stated in Chapter 2. The accurate solutions are not efficient; therefore, heuristic solutions can be used to find the near-optimal solutions. The bi-level optimization problem can be solved by bi-level nested genetic algorithm (BNGA). For BNGA, both the upper and lower level problems are solved by genetic algorithm. The upper level needs to obtain the optimal solution X using GA. The output X of the upper level is used as the input to the lower level to find the optimal solution Y also using GA. This output Y is as a feedback to the upper level. Then the upper level takes Y as input to get the updated solution X^*. This process is iterated until both the upper and lower level get the satisfactory solutions. This methodology is called BNGA since it has two levels of GA, and the lower level GA is nested into the upper level

GA. Each level optimizes one design variable and this design variable is taken as input to the other level.

4.4 Joint optimization of marketing-engineering

4.4.1 Upper level: Marketing

From the marketing perspective, the part worth utility of the k-th segment of the j -th product is U_{kj}. The part worth utility for each attribute level in the specific market segment is analyzed using conjoint analysis. U_{kj} is assumed as a linear function of the part worth utilities of each attribute level of product p_j, which is shown in equation 4-3 (Jiao and Zhang, 2005).

$$U_{kj} = \sum_{i=1}^{I} \sum_{l=1}^{L_k} \left(w_{ji} u_{kil} x_{jil} + \pi_j \right) + \varepsilon_{kj} \tag{4-3}$$

where u_{kil} is the part-worth utility for the l-th level of attribute a_i of the k-th segment. w_{ji} is the weights of attribute a_i contained in the product p_j. π_j is the constant related to the derivation of a composite utility with respect to product p_j. ε_{kj} is each segment-product pair's error term. x_{jil} is a binary design variable, $x_{jil} = 1$ if the l-th level of attribute a_i is chosen for product p_j, otherwise, $x_{jil} = 0$.

The customer preference model can also be generated from conjoint analysis. Customer's preference is modeled by relative preference and is shown in Equation 4-4 (Jiao and Zhang, 2005). The relative preference is U_{kj} of the overall utilities of all the products related to the market segment. The probabilistic choice rule used is the conditional multinomial logit choice rule (MNL).

$$D_{kj} = \frac{e^{\alpha U_{kj}}}{\sum_{n=1}^{N} e^{\alpha U_{kn}}} \tag{4-4}$$

where $\{U_{kn}\}_N$ is the associated deterministic utilities for all product alternatives considered in the choice set, which may also include competitor's products, N denotes the size of the choice set, and α is a scaling parameter.

4.4.2 Lower level: Engineering

The cost estimation is very difficult especially at the product line planning phase, since the details design of the product is not available at this point. At this level, only potential attributes are available to choose. In addition, design and manufacturing resources could be shared among multiple products in mass customization (Moore et al., 1999). Therefore, the traditional fixed costs and variable costs estimation are not suitable for product line planning.

In addition, the cost advantages in mass customization rely on the mass production efficiency. Therefore, it is more important to analyze the magnitudes of deviations from existing product and process platforms due to design changes and process variations in relation to product variety to justify optimal product offerings rather than the absolute amount of dollar costs (Tseng and Jiao, 1996). Therefore, Jiao and Tseng (2004) have proposed to model the cost of providing variety based on varying impacts on process capabilities. The process capability index is used to handle the sunk costs related to product line and shared resources. This research uses a pragmatic costing approach based on standard time estimation developed by Jiao and Tseng (1999) for engineering costs estimations. The idea of pragmatic costing approach is to allocate costs to established standard time. Therefore, there is no need to identify various cost drivers and cost-related activities. Therefore, for product line, the cost is estimated based on the part-worth standard time of different attribute levels. The cost of each product is estimated based on its expected cycle time. The expected cycle time is calculated by aggregating part-worth standard times of selected attributes. The expected cycle time can be used as a performance indicator of variation in process capabilities (Jiao and Tseng, 2004). The cycle time is used to justify the engineering cost, and thus the smaller it is the better. The cycle time can be described by normal distributions. Therefore, the one-side control limit specification capability index PCI using 3-sigma can be describe in Equation 4-5.

$$PCI = \frac{\mu^T - LSL^T}{3\sigma^T} \tag{4-5}$$

where μ^T is the mean of the estimated cycle time, σ^T is the standard deviation of the cycle time, and LSL^T is the lower specification limit. The LSL^T can be determined based on the best 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., 2004). PCI is large when the production process is easy to implement, and is small when the production process is difficult. A penalty function for a product, C_j,is introduced as the cost function based on the PCI which is shown in Equation 4-6.

$$C_j = \beta e^{\frac{1}{PCI_j}} = \beta e^{\frac{3\sigma_j^T}{\mu_j^T - LSL^T}} \tag{4-6}$$

where β is a constant to indicate the average dollar per variation of process capabilities. PCI_j is the respective process capability of the product p_j . μ_j^T and σ_j^T are the mean and standard deviation of the estimated cycle time for product p_j. LSL^T is the baseline of cycle times for all products which are produced within the process platform. The mean and standard deviation are calculated using Equation 4-7 and 4-8 respectively.

$$\mu_j^T = \sum_{i=1}^{I} \sum_{l=1}^{L_i} \left(\zeta_{ji} \mu_{il}^t x_{jil} + \omega_j \right) \tag{4-7}$$

$$\sigma_j^T = \sqrt{\sum_{i=1}^{I} \sum_{l=1}^{L_i} (\sigma_{il}^t x_{jil})^2} \tag{4-8}$$

where ζ_{ji} and ω_j are regression coefficient. μ_{il}^t and σ_{il}^t are the mean and standard deviation of the part-wort standard time of the l-th level of attribute a_i.

4.4.3 Joint optimization model

By comparing the upper level model and the lower level model, the joint optimization model is constructed as below:

$$\mathbf{Max_X} \sum_{k=1}^{K} \sum_{j=1}^{J} \boldsymbol{U_{kj} D_{kj} Q_k y_j} \tag{4-9.1}$$

$$\text{s.t.} \, U_{kj} = \sum_{i=1}^{I} \sum_{l=1}^{L_i} \left(w_{ji} u_{kil} x_{jil} + \pi_j \right) + \varepsilon_{kj} \,, \quad i \in \{1, \dots, I\}, j \in \{1, \dots, J\}, \;\; k \in \{1, \dots, K\} \tag{4-9.2}$$

$$D_{kj} = \frac{e^{\mu U_{kj}}}{\sum_{n=1}^{N} e^{\mu U_{kn}}}, \quad k \in \{1, \dots, K\}, j \in \{1, \dots, J\}, \tag{4-9.3}$$

$$\sum_{i=1}^{I} \sum_{l=1}^{L_i} x_{jil} = 1, \quad i \in \{1, \dots, I\}, j \in \{1, \dots, J\}, \tag{4-9.4}$$

$$\sum_{i=1}^{I} \sum_{l=1}^{L_i} \left| x_{jil} - x_{j'il} \right| > 0, \quad j, j' \in \{1, \dots, J\}, j \neq j', \tag{4-9.5}$$

$$\sum_{j=1}^{J} y_j \leq J^*, \;\; J^* \in \{1, \dots, J\}, \tag{4-9.6}$$

$$x_{jil}, y_j \in \{0,1\}, \; i \in \{1, \dots, I\}, j \in \{1, \dots, J\}, l \in \{1, \dots, L_i\}, \tag{4-9.7}$$

$$\mathbf{Min_Y} \sum_{k=1}^{K} \sum_{j=1}^{J} \boldsymbol{C_j Q_k y_j} \tag{4-9.8}$$

$$s.t. \, C_j = \beta e^{\frac{3\sigma_j^T}{\mu_j^T - LSL^T}}, \tag{4-9.9}$$

$$s.t. \; \mu_j^T = \sum_{i=1}^{I} \sum_{l=1}^{L_i} \left(\zeta_{ji} \mu_{il}^t x_{jil} + \omega_j \right), \quad i \in \{1, \dots, I\}, j \in \{1, \dots, J\}, \tag{4-9.10}$$

$$\sigma_j^T = \sqrt{\sum_{i=1}^{I} \sum_{l=1}^{L_i} (\sigma_{il}^t x_{jil})^2} \,, \; i \in \{1, \dots, I\}, j \in \{1, \dots, J\}, l \in \{1, \dots, L_i\}. \tag{4-9.11}$$

where Equation 4-9.1 and 4-9.8 show the objective functions for upper and lower level respectively. The design variable needs to be optimized for the upper level is X, which is a set of choices of each attribute level for potential products. The design variable needs to be optimized for the lower level is Y, which is a set of decisions of offering of each product. Equation 4-9.4 shows that each product must consist of at least one attribute. Equation 4-9.5 describes that two products cannot be the same in the product line. Equation 4-9.6 shows that the number of products in the product line cannot exceed the defined maximum number of products for the product line.

4.5 Chapter Summary

This chapter describes the methodology to solve the leader-follower joint optimization of product line evolution. The theoretic ground for the bi-level programming is illustrated with general solution step for the bi-level joint optimization. Then three types of leader-follower joint optimization are described. Both the accurate and heuristic solution strategies for bi-level joint optimization are discussed in this chapter. Then detailed upper level model to maximize the customer perceived value and lower level model to minimize engineering cost are presented. At last, the joint optimization model for the product line evolution problem which is the combination of the upper and lower level model is illustrated.

CHAPTER 5: BI-LEVEL NESTED GENETIC ALGORITHM (BNGA)

This chapter illustrates the development of bi-level nested genetic algorithm to solve the problem as formulated in Chapter 3 using the methodology developed in Chapter 4. Section 5.1 describes the rational to use BNGA solve the product line evolution problem. Section 5.2 shows the procedure of BNGA with a process flow chart. Section 5.3 explains the detailed software implementation of BNGA using MATLAB.

5.1 Rational of BNGA

Twelve attributes of three levels each may generate $3^{12} = 531441$ possible products. A product line consisting of five products may produce $(3^{12})^5 + (3^{12})^4 + (3^{12})^3 + (3^{12})^2 + (3^{12})^1 = 4.239 \times 10^{28}$ possible combinations. Therefore, it is not feasible to use enumeration to get the optimal product line. Complete enumeration to obtain optimal product selections in product line planning becomes numerically prohibitive (Tarasewich and Nair, 2001). In addition, product line planning problems are combinatorial optimization problems because typical attributes used are discrete (Kaul and Rao, 1995). Nearly all product line planning problems are NP-hard problem, therefore, heuristic solution strategies have been proposed to solve this type of problems (Nair et al., 1995). For the combinatorial optimization problem, it has been proven that genetic algorithm (GA) is excel comparing with traditional calculus based or approximation optimization techniques (Steiner and Hruschka, 2002).

GA uses a probabilistic technique based on the principle of natural selection. For product line planning, GA allows product profiles to be constructed directly from attribute level part-worth data (Kohli and Sukumar, 1990). Therefore, it can facilitate the one-step approach for the product line planning problem. In addition, the product line evolution problem cannot be solved using two-step approach. In other words, the product profiles for the potential product line must be constructed from attribute level part-worth data other than the reference set enumeration. Therefore, GA is adopted in this research to solve the joint optimization problem in Equation 4-9. Since this is a leader-follower game, a bi-level nested GA (BNGA) is developed. BNGA has two levels corresponding to the upper and lower levels. The lower level is nested into the upper level.

5.2 BNGA procedure

(1) Chromosome Coding: The first step to implement the BNGA is the encoding of the product and product line into chromosome. Figure 5-1 shows the GA composition for the upper and lower level of the model. For the upper level, the chromosome represents the potential products, and is also the design variable X, which indicates the choice of attribute levels for the potential products. For the lower level, the chromosome represents the product line, which is also the design variable Y, the decision of offerings of the potential products.

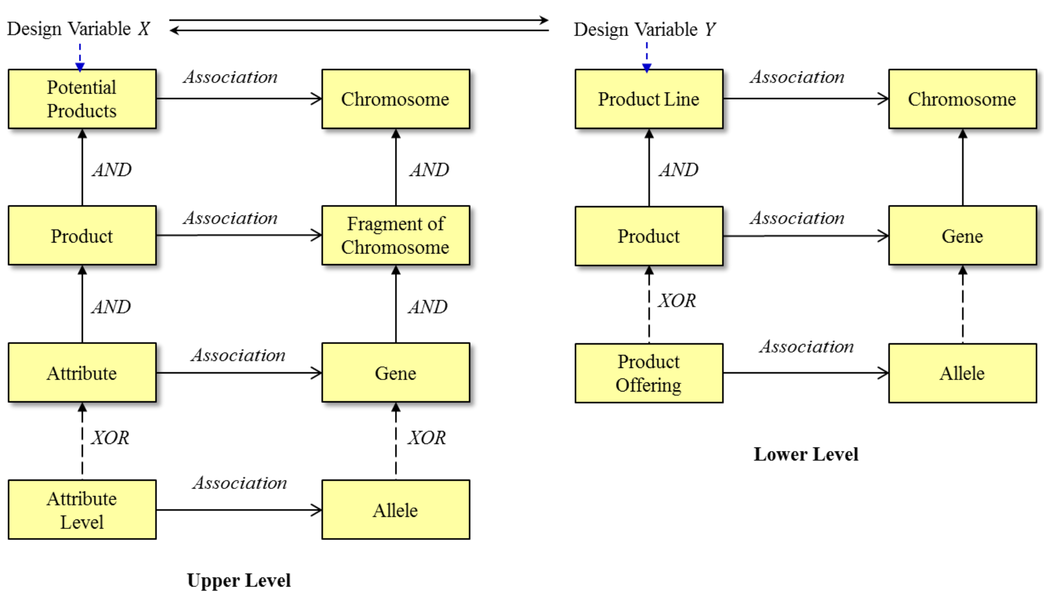

Figure 5-1. GA Composition

For one product, the chromosome contains I genes. Each gene represents an attribute. The value for the gene shows the attribute level. If the value is 0, this attribute is not selected for the respective product. Assume the product line will contain at most five products; a chromosome consisting of a 5-element string is coded. Each gene represents one product. If the value of the gene is 1, the product is chosen for the product line; otherwise, it is not selected. Figure 5-2 and Figure 5-3 show the detailed chromosome encoding for lower level and upper level respectively.

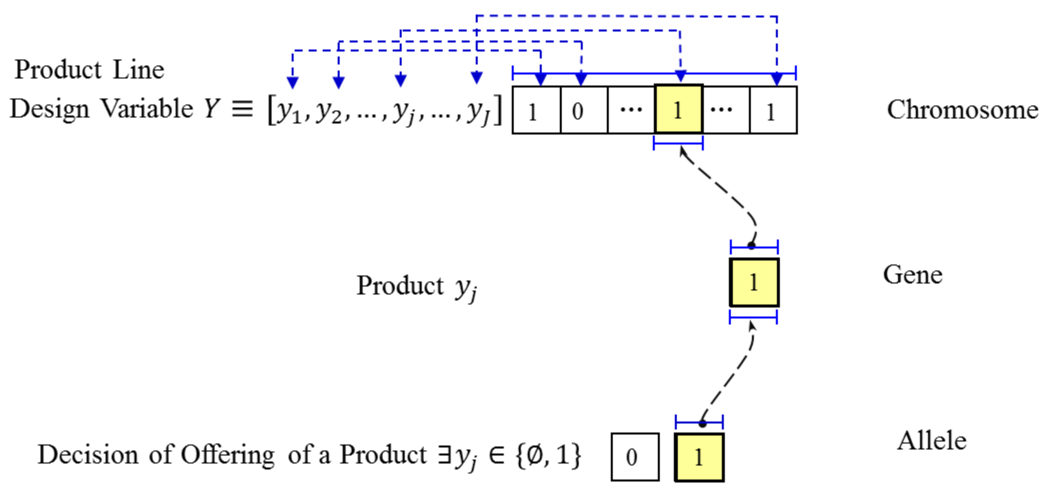

Figure 5-2. Chromosome Encoding for Lower Level

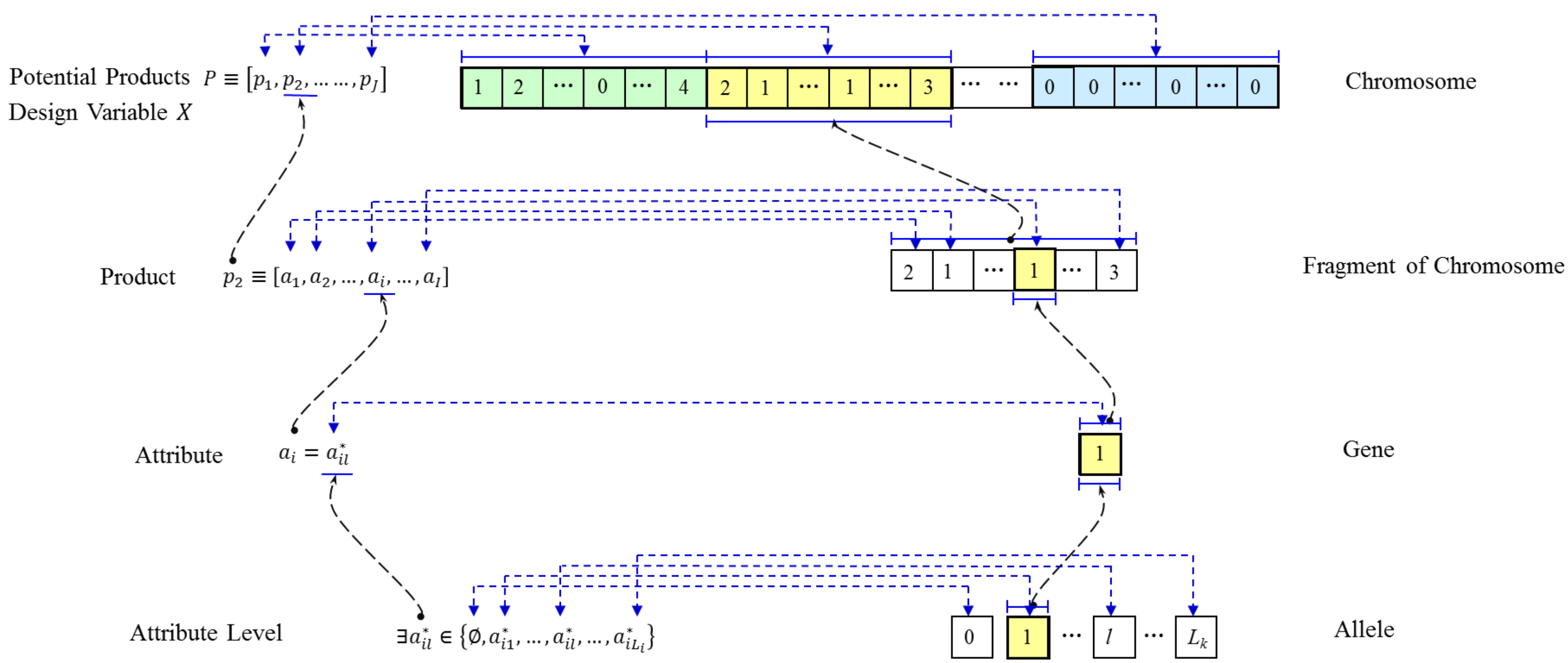

Figure 5-3. Chromosome Encoding for Upper Level

(2) Initialization: The initialization should generate initial solutions to the problem. The initial solution can be generated using random number generator (*rand* in MATLAB). The population size M determines number of chromosome strings encoded for each run. A larger number of M gives GA a larger chance of success since it can search in more solution space. M also affects the efficiency of the algorithm. A larger M will lead to more calculations and reduce the efficiency of GA.

(3) Fitness function: Fitness function is used to evaluate each individual chromosome within the population of each generation. The fitness function of the upper level model is Equation 4.9-1, of the lower level is Equation 4.9-8.

(4) Selection and reproduction: The fitness function is used to evaluate all generated chromosomes and then rank them in descending order. Only the top N chromosomes will be kept. The GA starts the parent selection and reproduction after the fitness function defined. Parent selection allocates reproductive opportunities for the chromosome among the population, which is based on the fitness value of individual chromosome.

(5) Crossover: Crossover is a genetic operator that can vary a chromosome from one generation to the next. After reproduction, each pair of two parent strings is randomly chosen and undergoes crossover with a probability. For each pair, the two individual chromosomes exchange their genetic composition to generate their offspring. Therefore, the offspring has some fragments of genes from each parent. This research uses a single point random crossover operator. The crossover operator randomly selects the cut-point. Then the offspring copies the first parent's genes from the start to the crossover point, and then inherits the second parent's genes from the crossover point to the end. The probability of crossover is defined by the crossover rate, which describes the percentage of chromosomes undergoing crossover in each generation.

(6) Mutation: Mutation is a genetic operator applied to each individual offspring after crossover, which is used to maintain and introduce genetic diversity from one generation to the next. It randomly picks a gene within each individual chromosome and alter the attribute level. The probability is defined by the mutation rate. The mutation rate of 0.001 is adopted, which is suggested that can produce good solutions from empirical findings (Gen and Cheng, 2000).

(7) Termination: The reproduction and crossover processes will not be terminated until the maximum iteration number is arrived. If the maximum iteration number is not reached, but the population has converged, the processes will also be terminated since the optimal solution has already been found. GA uses a moving average rule to indicate the convergence.

Figure 5-4 shows the process flow of BNGA. After the initialization process, the upper level GA runs to find the attribute choice in order to maximize the utility. Then, the termination criteria for the BNGA is checked. If the BNGA is not terminated, the updated attribute choices are input to lower level GA. Lower level GA runs to find the optimal decisions of product offerings to minimize engineering cost. The updated product choices are feedback to the upper level GA. Then the upper level GA runs again to find the new optimal attribute choices for potential products. This process iterates until it meets the termination criteria of the BNGA.

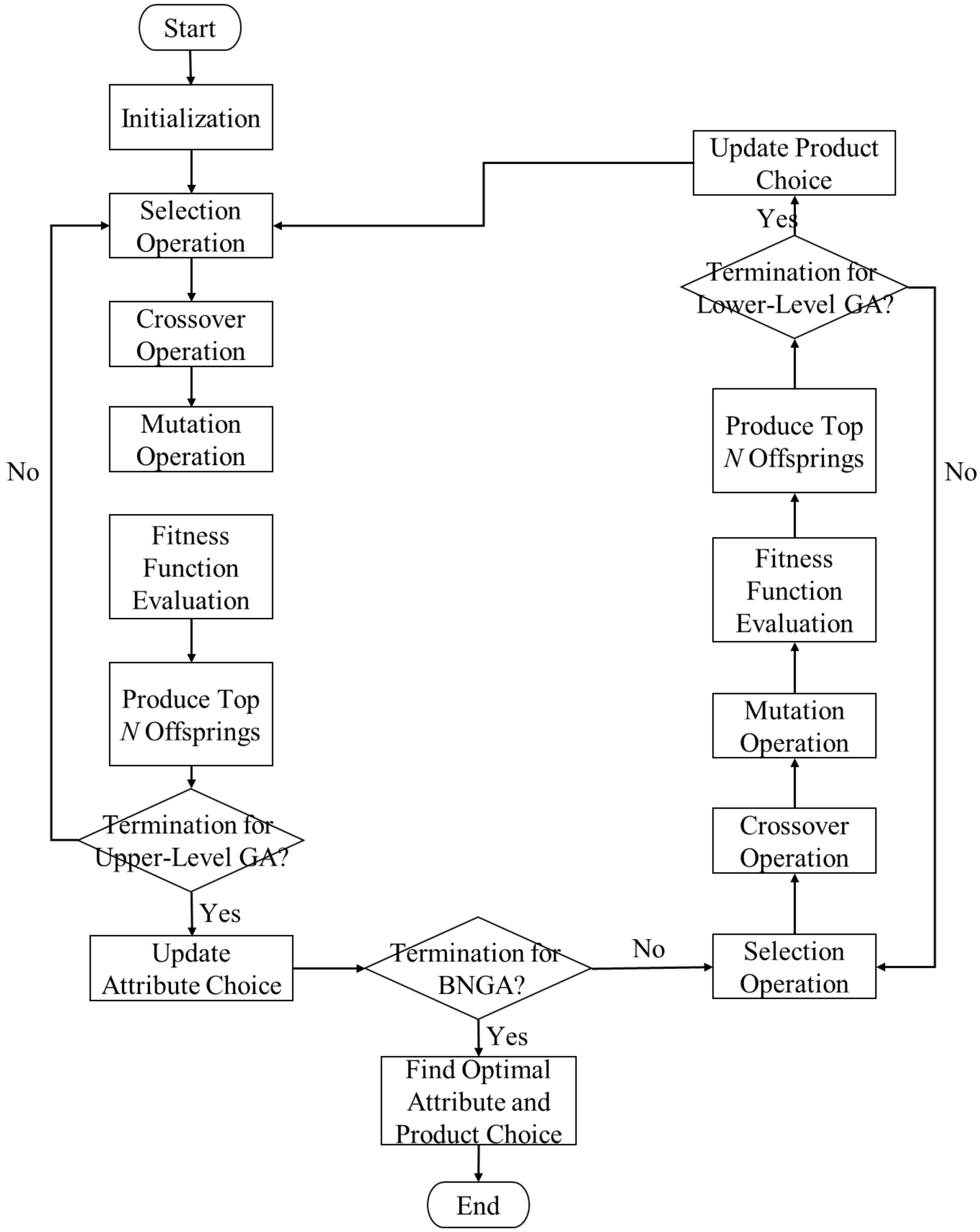

Figure 5-4. Process Flow of BNGA

5.3 Software implementation

This research adopts MATLAB as the software to implement the BNGA. The program is shown in the appendix. There are three functions defined in the program. The fitness function for upper level to maximize the customer perceived value, the fitness function for lower level to minimize the engineering cost, and the main function for BNGA which specifies the initialization, constraints and GA for both levels.

```
while e<n
    options_x = gaoptimset('PlotFcns',@gaplotbestf,'TolFun',1e-8,...
         'PopulationSize',1000,'PopInitRange',bound_x,'Generations',500,...
         'CrossoverFraction',0.55);
    FF_UL=@(x) UpperLevel(x,y);

[x,fval_x,exitflag_x]=ga(FF_UL,I*J,[],[],[],[],lb_x,ub_x,[],IntCon_x,options_x)

    utility=1./(UpperLevel(x,y));
    cost=LowerLevel(y,x);
    ratio=utility/cost;
    ratio_o=[ratio_o ratio];
    u_gen=[u_gen utility];
    c_gen=[c_gen cost];
    x_all=[x_all; x];
    y_all=[y_all; y];

    options_y = gaoptimset('PlotFcns',@gaplotbestf,'TolFun',1e-8,...
         'PopulationSize',1000,'PopInitRange',bound_x,'Generations',100,...
         'CrossoverFraction',0.55);
    FF_LL=@(y) LowerLevel(y,x);
    cons=@constraint;

[y,fval_y,exitflag_y]=ga(FF_LL,J,[],[],[],[],lb_y,ub_y,cons,IntCon_y,options_y)

    e=e+1;
end
```

Figure 5-5. MATLAB Program for BNGA

Figure 5-4 shows the iterations for BNGA in MATLAB. `options_x` and `options_y` shows the options for upper and lower level's GA respectively. `FF_UL` is the fitness function for upper level. It calls the upper level function `UpperLevel(x,y)`. `@(x)` means `x` is the variable which needs to be maximized for this fitness function. This program uses MATLAB function `ga` for genetic algorithm. `ga` is used to find the minimum of the function. Therefore, the upper level function calculates 1/utility. The first argument for `ga` function is the fitness function. The second one is the number of variables need to be maximized. `lb_x` , `ub_x`, `lb_y` and `ub_y` define the lower bound and upper bound for variables of both levels respectively. The initial input for the upper level is defined by the `rand` function in MATLAB. The output of the upper level, `x`, is the input for the lower level. Then the output of the lower level, `y`, is feedback to the upper level. This process will iterate for `n` generations. For each individual generation, the GA for upper level terminates when it runs for 500 generations or the change of mean of the fitness function is less than 10^{-8}; the GA for lower level terminates when it runs for 100 generations or the change of mean of the fitness function is less than 10^{-8}.

The utility, cost and utility/cost ratio for `n` generations of are recorded. Then the value and index for the maximum utility/cost ratio is found. The index is used to locate the corresponding `x` and `y`, which are the decisions of attributes offerings and product offerings.

5.4 Chapter Summary

This chapter first illustrates the rational of BNGA to solve the leader-follower joint optimization for product line evolution. The BNGA can solve the mixed integer combinatorial problem efficiently. Then the detailed procedure of BNGA is shown with a process flow chart. The last section explains the MATLAB implementation of BNGA.

CHAPTER 6: CASE STUDY OF SMART WATCH

In this chapter, a case study of smart watch is reported to illustrate the proposed methodology and solution. Section 6.1 introduces the case. Section 6.2 presents the results for the case study. Section 6.3 analyzes the results to validate the proposed approach, and also compares the bi-level joint optimization with all-in-one optimization to show the advantage of this method.

6.1 Problem context

Smart watch is a type of wearable device. It combines the functionality of traditional watch like timekeeping and some features of computers or mobile phone. A smart watch may include features such as touchscreen, camera, pedometer, etc. Compared to computers and mobile phones, smart watches are more advanced in the size, interfaces and especially service packages such as health related applications, E-payment service, etc. There are large amount features for smart watch. Therefore, it is necessary to select and find the optimal combination of these features to maximize the customer perceived value and minimize engineering cost. In addition, smart watch is relatively a new product. Hence, it evolves and updates very often. Therefore, product line evolution is very necessary for smart watch design. For the smart watch, feature is treated as attribute for product line evolution. Table 6-1 shows the attributes and their levels for smart watch.

Table 6-1. Attributes and Levels for Smart Watch

Attribute		Attribute Levels		
a_i	**Description**	a_{il}^*	**Code**	**Description**
a_1	Display shape	a_{11}^*	A1-1	Round
		a_{12}^*	A1-2	Square
a_2	Display size	a_{21}^*	A2-1	1.3 in^2
		a_{22}^*	A2-2	1.6 in^2
		a_{23}^*	A2-3	2.0 in^2
a_3	Internal memory	a_{31}^*	A3-1	1GB
		a_{32}^*	A3-2	4GB
		a_{33}^*	A3-3	8GB
a_4	Connectivity	a_{41}^*	A4-1	Wi-Fi
		a_{42}^*	A4-2	Wi-Fi & Cellular
		a_{43}^*	A4-3	GPS & Cellular
		a_{44}^*	A4-4	Wi-Fi, Cellular & GPS
a_5	Battery life	a_{51}^*	A5-1	< 20 hours
		a_{52}^*	A5-2	> 20 hours
a_6	Input device	a_{61}^*	A6-1	Touchscreen &microphone
		a_{62}^*	A6-2	Microphone & camera
		a_{63}^*	A6-3	Button & microphone
		a_{64}^*	A6-4	Touchscreen, microphone &camera
a_7	Output device	a_{71}^*	A7-1	Speaker
		a_{72}^*	A7-2	Speaker &IR blaster
		a_{73}^*	A7-3	Vibration/Haptic engine
a_8	Health & Fitness Sensor	a_{81}^*	A8-1	Pedometer
		a_{82}^*	A8-2	Pedometer & Heart rate sensor
		a_{83}^*	A8-3	Heart rate & SpO2 sensor
a_9	Environment sensor	a_{91}^*	A9-1	Light sensor
		a_{92}^*	A9-2	Digital compass & barometer
		a_{93}^*	A9-3	Gyrometer & light sensor
a_{10}	Operation System Compatibility	a_{101}^*	A10-1	Android
		a_{102}^*	A10-2	iOS
		a_{103}^*	A10-3	Android & iOS
a_{11}	Health & Fitness Service	a_{111}^*	A11-1	Exercise monitor
		a_{112}^*	A11-2	Exercise monitor & sleep monitor
		a_{113}^*	A11-3	Sleep monitor & family doctor
a_{12}	Utilities Software	a_{121}^*	A12-1	Notifications
		a_{122}^*	A12-2	Voice recognition & personal assistant
		a_{123}^*	A12-3	Notifications & media controller
		a_{124}^*	A12-4	Notifications & E-payment
		a_{125}^*	A12-5	Voice recognition , personal assistant & E-payment

There are in total 12 attributes and each attribute has several levels. These attributes are chosen based on research of existing products. These are the attributes that are different and provide variety of smart watch. These attributes are classified into three categories: a_1 to a_7 are traditional hardware; a_8 and a_9 are service related hardware; a_{10} to a_{12} are software. There are three market segments considered in this case study: the teenagers, adults and senior. The part-worth utilities for the attribute levels to each market segment and the part-worth standard time estimations are shown in the appendix.

6.2 Results

The part-worth utilities for the attribute levels to each market segment and the part-worth standard time estimations are inputs for the BNGA MATLAB program described in section 5.3. The product line results in highest utility/cost ratio is considered the optimal solution. Table 6-2 shows the optimal solution for the smart watch product line. It contains 3 products. Product 2 is a low-end product that serves as a traditional watch. Product 1 and 3 are both high-end products but with different types of output device, sensors and services, which can satisfy customer needs for different market segments.

Table 6-2. Optimal Solution for Smart Watch Product Line

Decision of attribute offerings	{1,2,3,4,2,4,3,3,2,3,3,5;2,1,1,1,2,0,3,0,0,1,0,0;2,3,1,3,1,0,0,0,0,1,0,0;2,2,1,1,1,0,0,0,0,1,0,0;1,3,1,3,2,0,0,0,0,1,0,0;1,2,3,4,2,4,2,2,2,3,2,5}		
Decision of product offerings	{1,1,0,0,0,1}		
	Product 1	Product 2	Product 3
Display shape	Round	Square	Round
Display size	1.6 in^2	1.3 in2	1.6 in^2
Internal memory	8 GB	1 GB	8 GB
Connectivity	Wi-Fi, Cellular & GPS	Wi-Fi	Wi-Fi, Cellular & GPS
Battery life	> 20 hours	> 20 hours	> 20 hours
Input device	Touchscreen, microphone &camera	Nil.	Touchscreen, microphone &camera
Output device	Vibration/Haptic engine	Vibration/Haptic engine	Speaker &IR blaster
Health & fitness sensor	Heart rate & SpO2 sensor	Nil.	Pedometer & Heart rate sensor
Environment sensor	Digital compass & barometer	Nil.	Digital compass & barometer
OS compatibility	Android & iOS	Android	Android & iOS
Health & fitness service	Sleep monitor & family doctor	Nil.	Exercise monitor & sleep monitor
Utilities software	Voice recognition , personal assistant & E-payment	Nil.	Voice recognition , personal assistant & E-payment

6.3 Analysis

For BNGA, the upper level minimizes 1/utility, and the lower level minimizes the engineering cost. For each individual iteration, the best value of the fitness function and mean value of the fitness function for both levels look like Figure 6-1 and 6-2 respectively.

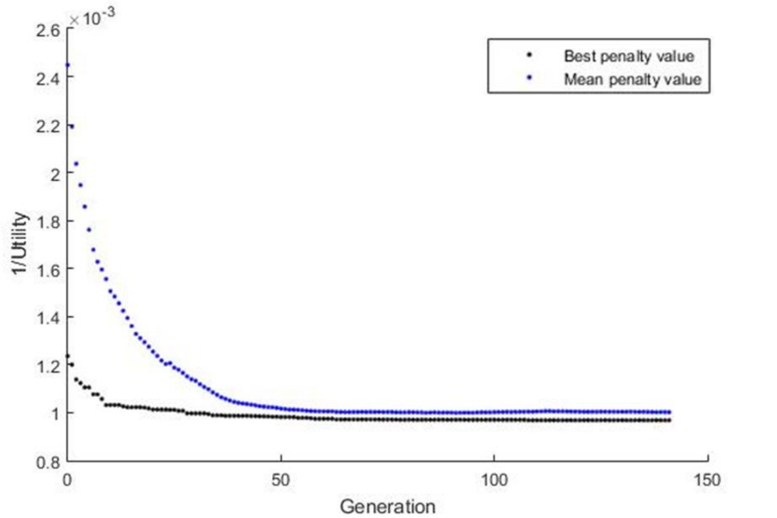

Figure 6-1. 1/Utility (Upper Level) for An Individual Iteration using GA

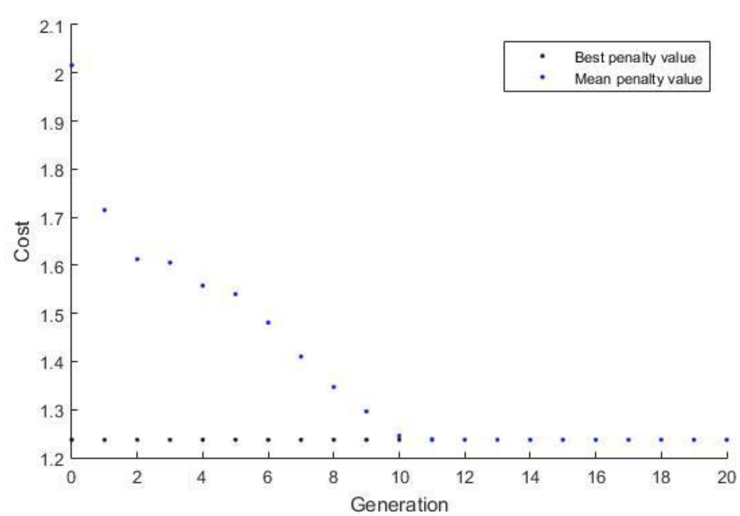

Figure 6-2. Cost (Lower Level) for An Individual Iteration using GA

Figure 6-3, 6-4 and 6-5 shows the utilities, costs and utility/cost ratios among the 500 iterations. The maximum utility/cost ratio occurs at the 430th iteration. For the 430th generation, the utility is 1037, the cost is 1.238, and the utility/cost ratio is 837.1.

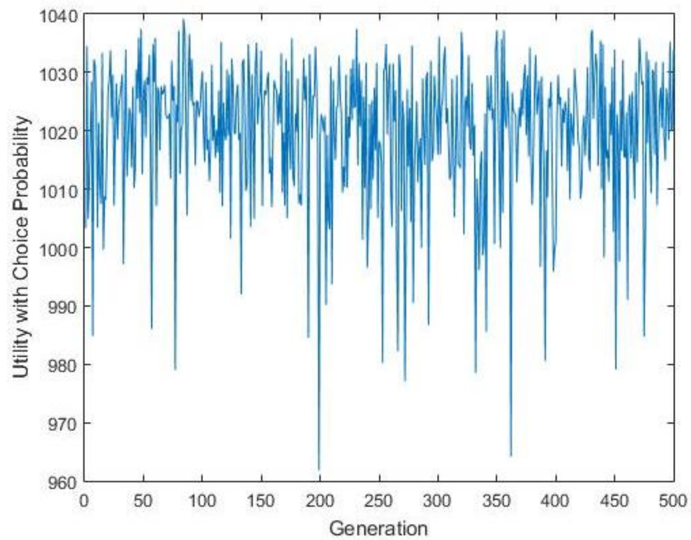

Figure 6-3. Utility among 500 Iterations

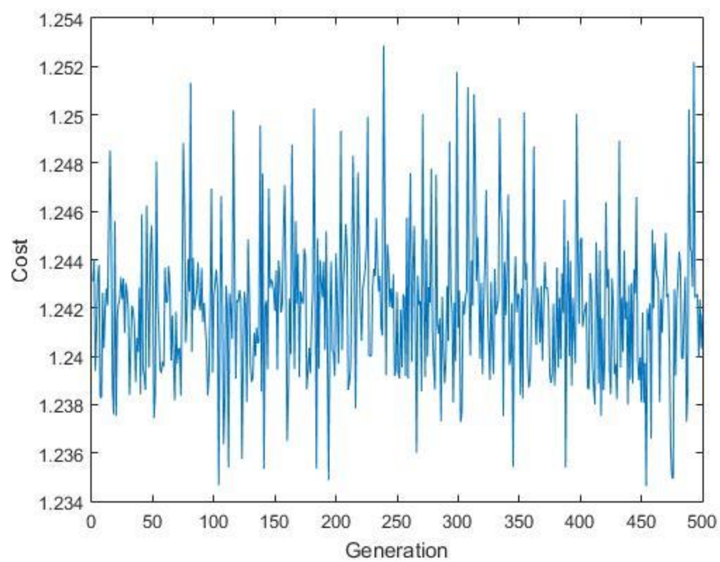

Figure 6-4. Cost among 500 Iterations

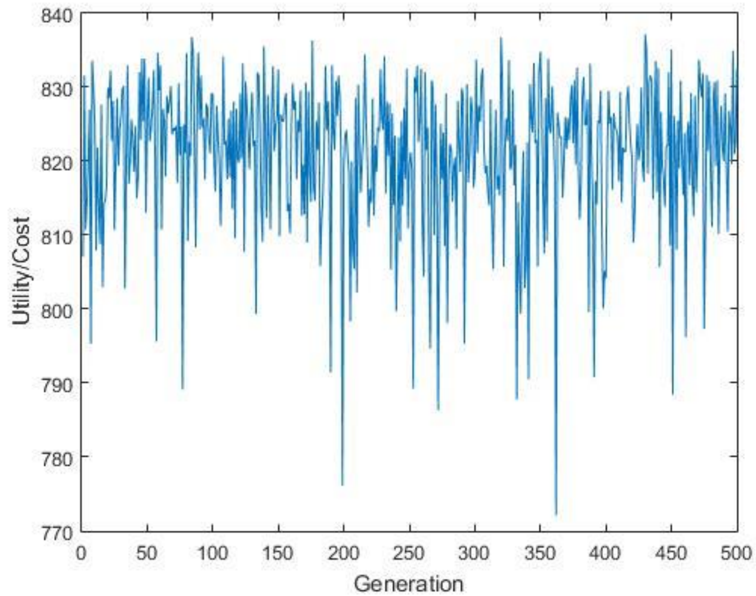

Figure 6-5. Utility/Cost Ratio among 500 Iterations

Table 6-3 and Figure 6-6 show the comparison of maximum ratio, maximum utility and minimum cost points among the 500 generations. Among the 500 generations, the maximum utility occurs at the 84^{th} generation. At this generation, the cost is more compared to the cost of the 430^{th} generation, and the cost/utility ratio is less than that of the 430^{th} generation. The minimum cost occurs at the 454^{th} generation.

Table 6-3 Comparison of Maximum Ratio, Maximum Utility and Minimum Cost

	Generation	Cost/Utility Ratio	Utility	Cost
max(Cost/Utility Ratio)	430	837.1	1037	1.238
max(Utility)	84	836.8	1040	1.242
min(Cost)	454	808.0	997.6	1.235

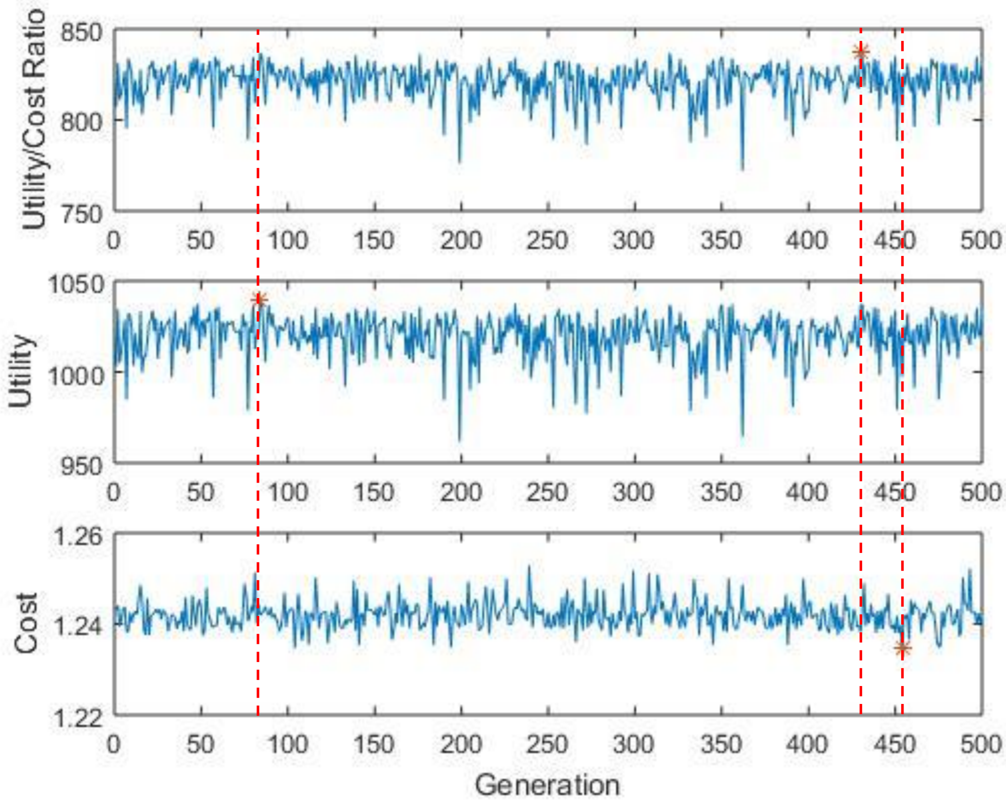

Figure 6-6. Comparison of Maximum Ratio, Maximum Utility and Minimum Cost

A comparison experiment is tested using the shared-surplus method (Jiao and Zhang, 2005). This is an all-in-one method. The cost/utility ratio among the process of GA is shown in Figure 6-6. For this experiment, the optimal utility/cost ratio is 419.8. Therefore, the bi-level joint optimization using BNGA can produce double utility/cost ratio compared to the all-in-one method. The all-in-one method is more efficient than bi-level joint optimization since this method only need to run single level GA once. The proposed approach need to run the upper level GA and lower level GA for 500 iterations. For each iteration, the upper level GA will get the optimal results in around 150 generations, and the lower level GA will get the optimal solution in around 20 generations. Although the bi-level joint optimization model is less efficient, it can get much better results than the all-in-one method. Therefore, it is worth the cost of efficiency.

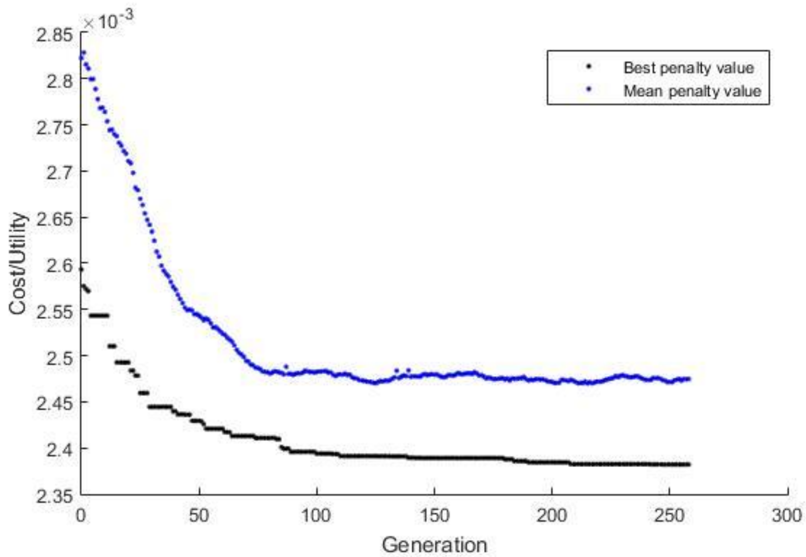

Figure 6-7. Cost/Utility using All-In-One Method

6.4 Chapter Summary

This chapter presents a case study of smart watch. The problem context is first illustrated. Then the optimal solution for the smart watch product line is presented. The optimal product line consists of three products to maximize the utility/cost ratio. Then the results are analyzed to show that the proposed approach leveraging marketing and engineering perspective is better than only using the upper level or lower level model. Also, an all-in-one method which combines the utility and cost in one equation and use only one level of GA is used to find the optimal solutions. The optimal solutions derived from all-in-one methods can only produce half of the utility/cost ratio derived from the bi-level joint optimization model.

CHAPTER 7: CONCLUSIONS

This chapter concludes this thesis in three sections. The first section presents the contributions of this thesis. Section 7.2 shows the limitations of the research. Finally, section 7.3 briefly discusses the work could be done in the future based on current research.

7.1 Contribution

This research illustrates the new idea of product line evolution. Product line evolution is originated from the product line planning problem. It updates the product line based on existing product lines. It can also serve for future generations. Then this thesis proposes a bi-level joint optimization model for product line evolution. Compared to the traditional all-in-one model to product line evolution problem, joint optimization model can leverage the conflict goals of marketing perspective and engineering perspective.

Product line evolution is a combinatorial optimization problem. Hence, a bi-level genetic algorithm is developed and applied to this combinatorial optimization problem. BNGA is composed of two individual GA operations for the upper level model and lower level model. The lower level GA is nested into the upper level GA. BNGA is used to find the optimal solutions for the attribute choice and the product choice.

A case study of smart watch product line evolution is reported to illustrate the feasibility and potential of the proposed approach. Compared to all-in-one method, the bi-level optimization model could generate double utility/cost ratio.

7.2 Limitations

There are still some limitations in this research. This thesis only focuses on the optimization stage of the product line evolution and does not propose new methodology for the product line generation stage. The current case study addresses the traditional static optimization for the product line evolution. It does not show the dynamic process of the product line evolution. For the evolution purpose, the proposed approach can only select product attributes from what manufacturers already have, rather than creating new attributes. Hence, it cannot innovate new ideas for the product designer. It is only able to assist the company to construct products from available attributes. Moreover, this research does not consider competitor's product. In addition, factors such as the new technologies, the releasing time of a new product which will influence the customer's preferences and the perceived value are not considered in this research.

In addition, BNGA is a stochastic method. Therefore, it can only find the near-optimal solutions to the problem, but cannot guarantee to find the best solutions.

7.3 Future work

Based on the current computational model and approach, more work can be done to make this research more complete. An improved method could be developed for the product line generation stage in order to get a better product line evolution model. The competitors' products should be involved into the model. Since most companies will react to their competitors' new product lines, the competition model should be developed to simulate the dynamic competition among the markets. For the case study, it only considers a small number of available attributes. However, the customers may consider many more factors when purchasing a product. In the future, a more comprehensive case study should be done to cover more attributes. In addition, future case study should also consist the dynamic optimization process for the product line evolution problem. Moreover, this case study only considers three market segments. In reality, the markets are highly segmented due to various reasons such as geographical ones.

APPENDIX A: PART-WORTH DATA

Table A-1. Part-Worth Preference and Standard Time Table

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	0.92	0.81	0.69	436	10.8
A1-2	0.62	0.70	0.75	325	8.5
A2-1	0.63	0.73	0.62	232	4.1
A2-2	0.84	0.68	0.86	298	4.6
A2-3	0.56	0.80	0.98	336	5.1
A3-1	0.52	0.54	0.64	648	6.2
A3-2	0.86	0.65	0.73	683	6.4
A3-3	0.94	0.78	0.71	725	6.6
A4-1	0.83	0.65	0.76	278	4.4
A4-2	0.71	0.93	0.72	392	9.6
A4-3	0.68	0.84	0.68	416	10.3
A4-4	0.92	0.99	0.84	694	12.4
A5-1	0.72	0.64	0.52	184	3.6
A5-2	0.83	0.93	0.98	281	4.2
A6-1	0.94	0.85	0.72	428	9.8
A6-2	0.89	0.89	0.76	686	16.3
A6-3	0.62	0.81	0.82	317	7.4
A6-4	0.97	0.91	0.79	833	26
A7-1	0.88	0.84	0.94	196	3.7
A7-2	0.96	0.89	0.86	361	6.8
A7-3	0.93	0.92	0.81	249	4.4
A8-1	0.76	0.71	0.84	207	3.8
A8-2	0.91	0.79	0.92	529	12.1
A8-3	0.78	0.93	0.99	672	16.3
A9-1	0.71	0.62	0.53	198	3.7
A9-2	0.82	0.95	0.98	273	4.9
A9-3	0.79	0.73	0.67	365	7.0
A10-1	0.64	0.79	0.54	98	10.2
A10-2	0.86	0.88	0.57	105	13.5
A10-3	0.95	0.97	0.79	182	20.6

Table A-1 (continued).

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)**
A11-1	0.59	0.66	0.73	79	9.8
A11-2	0.73	0.81	0.88	114	16.8
A11-3	0.70	0.86	0.95	237	23.4
A12-1	0.72	0.66	0.55	51	5.6
A12-2	0.79	0.83	0.60	226	26.8
A12-3	0.84	0.70	0.57	139	17.3
A12-4	0.82	0.81	0.58	154	18.2
A12-5	0.97	0.89	0.73	329	35.9

APPENDIX B: MATLAB CODES FOR BNGA

B-1 Main function

```
function [x_all,y_all,ratio_o,u_gen,c_gen] = bnga(n)
%%Initialization
J=6;
IntCon_x=1:(12*J);
lb=[1 1 1 1 1 0 0 0 0 1 0 0];
ub=[2 3 3 4 2 4 3 3 3 3 3 5];
ub_x=[];
lb_x=[];

for i=1:J
    ub_x=[ub_x ub];
    lb_x=[lb_x lb];
end

bound_x=[lb_x;ub_x];

IntCon_y=1:J;
lb_y=zeros(1,J);
ub_y=ones(1,J);
bound_y=[lb_y;ub_y];


function [c ceq]=constraint(y)
c=[-(sum(y)-3)];
ceq=[];
end

y=round(rand(1,J));
e=0;
ratio_o=[];
u_gen=[];
c_gen=[];
x_all=[];
y_all=[];
while e<n
    options_x = gaoptimset('PlotFcns',@gaplotbestf,'TolFun',1e-
8,'PopulationSize',1000,...
        'PopInitRange',bound_x,'Generations',500,'CrossoverFraction',0.55,...
        'MigrationFraction',0.55);
    FF_UL=@(x) UpperLevel(x,y);

[x,fval_x,exitflag_x]=ga(FF_UL,12*J,[],[],[],[],lb_x,ub_x,[],IntCon_x,options
_x)
    utility=1./(UpperLevel(x,y));
    cost=LowerLevel(y,x);
    ratio=utility/cost;
    ratio_o=[ratio_o ratio];
    u_gen=[u_gen utility];
    c_gen=[c_gen cost];
    x_all=[x_all; x];
```

```
    y_all=[y_all; y];

    options_y = gaoptimset('PlotFcns',@gaplotbestf,'TolFun',1e-
8,'PopulationSize',1000,...
        'PopInitRange',bound_y,'Generations',100,'CrossoverFraction',0.55,...
        'MigrationFraction',0.55);
FF_LL=@(y) LowerLevel(y,x);
cons=@constraint;
[y,fval_y,exitflag_y]=ga(FF_LL,J,[],[],[],[],lb_y,ub_y,cons,IntCon_y,options_
y)

    e=e+1;
end
end
```

B-2 Upper level function

```
function UL=UpperLevel(x,y)
J = 6;
K = 3;
U(:,:,1)=[0,0.92, 0.62,0,0,0;...
    0,0.63,0.84,0.56,0,0;...
    0,0.52,0.86,0.94,0,0;...
    0,0.83,0.71,0.68,0.92,0;...
    0,0.72,0.83,0,0,0;...
    0,0.94,0.89,0.62,0.97,0;...
    0,0.88,0.96,0.93,0,0;...
    0,0.76,0.91,0.78,0,0;...
    0,0.71,0.82,0.79,0,0;...
    0,0.64,0.86,0.95,0,0;...
    0,0.59,0.73,0.70,0,0;...
    0,0.72,0.79,0.84,0.82,0.97];
U(:,:,2)=[0,0.81,0.70,0,0,0;...
    0,0.73,0.68,0.80,0,0;...
    0,0.54,0.65,0.78,0,0;...
    0,0.65,0.93,0.84,0.99,0;...
    0,0.64,0.63,0,0,0;...
    0,0.85,0.89,0.81,0.91,0;...
    0,0.84,0.89,0.92,0,0;...
    0,0.71,0.79,0.93,0,0;...
    0,0.62,0.95,0.73,0,0;...
    0,0.79,0.88,0.97,0,0;...
    0,0.66,0.81,0.86,0,0;...
    0,0.66,0.83,0.70,0.81,0.89];
U(:,:,3)=[0,0.69,0.75,0,0,0;...
    0,0.62,0.86,0.98,0,0;...
    0,0.64,0.73,0.71,0,0;...
    0,0.76,0.72,0.68,0.84,0;...
    0,0.52,0.98,0,0,0;...
    0,0.72,0.76,0.82,0.79,0;...
    0,0.94,0.86,0.81,0,0;...
    0,0.84,0.92,0.99,0,0;...
    0,0.53,0.98,0.67,0,0;...
    0,0.54,0.57,0.79,0,0;...
    0,0.73,0.88,0.95,0,0;...
```

```
    0,0.55,0.60,0.57,0.58,0.73];
Q=[50 20 30];
max_obj = 0;
for j=1:J
    x_in(j,:)=x(1+12*(j-1):12+12*(j-1));
end
for k = 1:K
    for j = 1:J
        Ukj(k,j)  =  U(1,x_in(j,1)+1,k)+U(2,x_in(j,2)+1,k)+U(3,x_in(j,3)+1,k)...
            +U(4,x_in(j,4)+1,k)+U(5,x_in(j,5)+1,k)+U(6,x_in(j,6)+1,k)...
            +U(7,x_in(j,7)+1,k)+U(8,x_in(j,8)+1,k)+U(9,x_in(j,9)+1,k)...
            +U(10,x_in(j,10)+1,k)+U(11,x_in(j,11)+1,k)+U(12,x_in(j,12)+1,k);
        eUkj(k,j)=exp(Ukj(k,j));
    end
end

denominator=sum(eUkj');
for k=1:K
    for j=1:J
        Pkj(k,j)=eUkj(k,j)/denominator(k);
    end
end

for k=1:K
    for j=1:J
        max_obj = max_obj + Ukj(k,j) * Pkj(k,j) * Q(k) * y(j);
    end
end

UL=1/max_obj;
end
```

B-3 Lower level function

```
function LL=LowerLevel(y,x)
I = 50;
J = 6;
K = 3;
LSL=45;
beta=0.004;
mu=[0,436,325,0,0,0;...
    0,232,298,336,0,0;...
    0,648,683,725,0,0;...
    0,278,392,416,694,0;...
    0,184,281,0,0,0;...
    0,428,686,317,833,0;...
    0,196,361,249,0,0;...
    0,207,529,672,0,0;...
    0,198,273,365,0,0;...
    0,98,105,182,0,0;...
    0,79,114,237,0,0;...
    0,51,226,139,154,329];
sigma=[0,10.8,8.5,0,0,0;...
    0,4.1,4.6,5.1,0,0;...
    0,6.2,6.4,6.6,0,0;...
```

```
    0,4.4,9.6,10.3,12.4,0;...
    0,3.6,4.2,0,0,0;...
    0,9.8,16.3,7.4,26,0;...
    0,3.7,6.8,4.4,0,0;...
    0,3.8,12.1,16.,0,0;...
    0,3.7,4.9,7.0,0,0;...
    0,10.2,13.5,20.6,0,0;...
    0,9.8,16.8,23.4,0,0;...
    0,5.6,26.8,17.3,18.2,35.9];
Q=[50 20 30];
min_obj = 0;
for j=1:J
    x_in(j,:)=x(1+12*(j-1):12+12*(j-1));
end
for k = 1:K
    for j = 1:J
        muj(j) = mu(1,x_in(j,1)+1)+mu(2,x_in(j,2)+1)+mu(3,x_in(j,3)+1)...
            +mu(4,x_in(j,4)+1)+mu(5,x_in(j,5)+1)+mu(6,x_in(j,6)+1)...
            +mu(7,x_in(j,7)+1)+mu(8,x_in(j,8)+1)+mu(9,x_in(j,9)+1)...
            +mu(10,x_in(j,10)+1)+mu(11,x_in(j,11)+1)+mu(12,x_in(j,12)+1);
        sigmaj(j) =
sqrt(sigma(1,x_in(j,1)+1)^2+sigma(2,x_in(j,2)+1)^2+sigma(3,x_in(j,3)+1)^2 ...

+sigma(4,x_in(j,4)+1)^2+sigma(5,x_in(j,5)+1)^2+sigma(6,x_in(j,6)+1)^2 ...

+sigma(7,x_in(j,7)+1)^2+sigma(8,x_in(j,8)+1)^2+sigma(9,x_in(j,9)+1)^2 ...

+sigma(10,x_in(j,10)+1)^2+sigma(11,x_in(j,11)+1)^2+sigma(12,x_in(j,12)+1)^2  );
        C(j)=beta*exp(3*sigmaj(j)/(muj(j)-LSL));
    end
end
for k=1:K
    for j=1:J
        min_obj = min_obj + C(j) * Q(k) * y(j);
    end
end
LL=min_obj;
end
```

REFERENCES

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

Bard, J.F., 1998. Practical Bilevel Optimization: Algorithms and Applications. Kluwer Academic Publishers, Dordrecht.

Brands, T., van Berkum, E.C., 2014. Performance of a Genetic Algorithm for Solving the Multi-Objective, Multimodal Transportation Network Design Problem International Journal of Transportation 2, 1-20.

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

Bryan, A., Ko, J., Hu, S. J., & Koren, Y. (2007). Co-evolution of product families and assembly systems. CIRP Annals-Manufacturing Technology, 56(1), 41-44.

Calvete, H.I., Galé, C., Mateo, P.M., 2008. A new approach for solving linear bilevel problems using genetic algorithms. European Journal of Operational Research 188, 14-28.

Chen, K.D., 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.

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

Colson, B., Marcotte, P., Savard, G., 2007. An overview of bilevel optimization. Ann Oper Res 153, 235-256.

Du, G., Jiao, R.J., Chen, M., 2014. Joint optimization of product family configuration and scaling design by Stackelberg game. European Journal of Operational Research 232, 330-341.

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

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

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

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

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

Hingston, P. F., Barone, L. C., and Michalewicz, Z. (2008). Design by evolution: advances in evolutionary design, Springer.

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

Jiao, J., Tseng, M.M., 2004, Customizability analysis in design for mass customization, *Computer-Aided Design*, 36(8): 747-757.

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

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

Ji, Y., Jiao, R.J., Chen, L., Wu, C., 2013. Green modular design for material efficiency: a leader-follower joint optimization model based on constrained genetic algorithm. Journal of Cleaner Production 41, 187-201.

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

Kim, K., 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.

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

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

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

Krishnan, V., 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.

Lai, Y.-J., 1996. Hierarchical optimization: A satisfactory solution. Fuzzy Sets and Systems 77, 321-335.

Li, H., 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.

Li, H., Zhang, L., Jiao, Y., 2014. Solution for integer linear bilevel programming problems using orthogonal genetic algorithm. Journal of Systems Engineering and Electronics 25, 443-451.

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

Monroe, K., Sunder, S., Wells, W.A., 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.

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

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

Otto, K. N. and K. L. Wood (1998). "Product evolution: a reverse engineering and redesign methodology." Research in Engineering Design 10(4): 226-243.

Pine, B.J.,1993, *Mass Customization: The New Frontier in Business Competition*, Harvard Business Review, 71(5): 108-121.

Pine, B.J., Victor, B., Boynton, A.C., 1993, *Making mass customization work*, Harvard Business Review, 71(5): 108-121.

Pullmana, M.E., Mooreb, W.L., Wardellb, 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.

Ramdas, K., Sawhney, M.S., 2001, A cross-functional approach to evaluating multiple line extensions for assembled products, *Management Science,* 47(1): 22-36.

Saari, D.G., 2000, Mathematical structure of voting paradoxes: I. Pairwise votes, *Economic Theory*, 15(1): 1-53.

Stackelberg, H., 1952. The Theory of Market Economy. Oxford University Press, Oxford, UK.

Steiner, W.J., 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.

Tseng, M.M. and Jiao, J., 1996, Design for mass customization, *CIRP Annals*, 45(1): 153-156.

Tarasewich, P., Nair, S.K., 2001, Designer-moderated product design, *IEEE Transactions on Engineering Management*, 48(2): 175-188.

Tellis, G. J. and C. Merle Crawford (1981). "An Evolutionary Approach to Product Growth Theory." Journal of Marketing 45(4): 125-132.

Wassenaar, H.J., Chen, W., 2001, An approach to decision-based design, Proceedings ASME 2001 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Pittsburgh, Pennsylvania, DETC2001/DTM-21683.

Wortmann, J.C., Muntslag, D.R., Timmermans, P.J.M., 1997, *Customer Driven Manufacturing*, Chapman & Hall, London.

Yano, C., 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.), Kluwer Academic Publisher, pp. 145-176.