European Journal of Operational Research 262 (2017) 207–216

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Decision Support

Joint decisions on product line selection, purchasing, and pricing

Ilkyeong Moon[a], Kun Soo Park[b,*], Jing Hao[c], Dongwook Kim[d]

[a] Department of Industrial Engineering and Institute for Industrial Systems Innovation, Seoul National University, Seoul, 08826, South Korea
[b] College of Business, Korea Advanced Institute of Science and Technology (KAIST), 85 Hoegi-ro, Dongdaemun-gu, Seoul, 02455, South Korea
[c] Lenovo, China
[d] Department of Industrial Engineering, Seoul National University, Seoul, South Korea

A R T I C L E I N F O

Article history:
Received 28 January 2016
Accepted 24 March 2017
Available online 31 March 2017

Keywords:
Product pricing
Product line selection
Dynamic substitution
Hybrid genetic algorithm

A B S T R A C T

When creating a product line, a retailer must make several decisions simultaneously: the selection of the product types to include in the product line as well as the order quantity and price of each selected product type. This study investigates joint product line decisions by considering the dynamic substitutions of products as driven by the valuations that customers place on the products and the availability of each product type, which changes as consumers purchase a product. An integer programming model was developed for joint decisions of product selection, price, and order quantity in the product line problem to maximize the total profit of the retailer. To solve the model, we propose a hybrid genetic algorithm (HGA) that uses special genetic operators and heuristic algorithms to ensure the feasibility and efficiency of solutions. Computational experiments demonstrate the superiority of the proposed HGA over the solution obtained by CPLEX for large scale problems. Useful managerial insights on the joint product line decisions are also derived from the numerical results.

1. Introduction

A retailer's product line refers to a set of substitutable goods that serve identical needs or markets but differ in some secondary aspects. A product line may be composed of the same products with different brands or brand products that differ in terms of color, function, origin, or flavor (Maddah & Bish, 2007). Retailers manage their product lines by adding or removing a type of product in their stores to satisfy customers from different markets through careful selection of prices and order quantities.

Retailers usually prefer to keep a steady product line because consumers find shopping easier if they can compare products during sales periods. To attract customers from many market segments, a retailer broadens the product line by including more product types. However, because shelf and storage spaces are limited, the retailer may need to include a limited number of product types, and therefore, customers may fail to find the product type they want. Furthermore, managing different types of products may increase operational costs, such as those related to purchasing, holding inventory, and conducting administrative functions, which harm the retailer's profit. Thus, the retailer should carefully decide which products are included in the product line, and many use optimization tools in this situation (Irion, Lu, Al-Khayyal, & Tsao, 2012).

The product line decision should be made concurrently with other relevant operational decisions, such as pricing and purchasing. These decisions help determine the inventory level of each product in the selected product line. Through the collection and analysis of sales data, retailers can determine the price and order quantity of each product type in the product line while considering the preferences of customers who belong to different market segments. Hence, this study addresses the joint product line decisions problem with consideration of pricing and purchasing decisions as based on customer preferences for product types.

In a retail store, consumers place value on each type of product and choose the product types that maximize their surplus; that is, they select a product after they determine the difference between the valuation they place on the product (reservation price) and the price the retailer places on it. This choice behavior is referred to as the *max-surplus choice rule* (MSCR) in the literature; see, for example, Green and Krieger (1985). In the product line decisions problem, the MCSR is an often-cited deterministic rule that describes the customer's choice of a product type in the product line based on the highest consumer surplus (Mayer & Steinhardt, 2016).

If the preferred choice of product type is not available, the customer may switch to another product type that is available and can be substituted for the favorite one. Because the availability of products is successively reduced by a customer's purchase,

* Corresponding author.
E-mail addresses: kunsoo@kaist.ac.kr, kunsoo.park@gmail.com (K.S. Park).

http://dx.doi.org/10.1016/j.ejor.2017.03.062
0377-2217/© 2017 Elsevier B.V. All rights reserved.

buyers who come later face fewer available products in the store than those who came earlier. As a result, late customers may buy products that they would not select if their first choices had been available. We consider the retailer's decision under such customer substitution behavior, which is called *dynamic substitution*. See, e.g., Mahajan and van Ryzin (2001) and Burkart, Klein, and Mayer (2012) for the details. Despite the popular use of dynamic substitution in retail stores, few researchers have studied it in relation to the product line decisions problem.

Product line selections as well as purchasing and pricing decisions are important and have significant impacts on retailers' profits. Due to the complex interrelationship among these choices and the large scale of the problem, jointly considering these three decisions is challenging; however, the solution is important in light of growing interests on the product line decisions problem. The objective of this study is to develop a decision model that can help a retailer simultaneously determine the best product selection, pricing, and purchasing decisions for a product line. An integer programming (IP) model is used to describe the problem and a hybrid genetic algorithm (HGA) is proposed to efficiently solve it. The analysis shows that the proposed algorithm successfully computes the near-optimal solutions reasonably well.

This paper is organized as follows: in Section 2, the literature review is presented and in Section 3, an IP model is developed for the joint product line decisions problem. In Section 4, a hybrid genetic algorithm is used to solve the realistic, large, product line selection problem. In Section 5, computational experiments are conducted, and in Section 6, conclusions are offered.

2. Literature review

Monroe et al. (1976) developed one of the first integer programming models for the product line selection problem by maximizing the discount cash flows from the selected product line. Later, Shugan and Balachandran (1977) and Dobson and Kalish (1988) incorporated choices of consumers, who aim to maximize their utilities in their product line problem. Zufryden (1982) introduced a binary integer program for the product line selection problem. The objective was to maximize the sum of consumers' choices under the max-surplus choice rule (MSCR). Green and Krieger (1985) proposed a heuristic algorithm for the product line selection problem by considering the total utility measure of the retailer's optimal choice of product line that incorporates the consumers' optimal choice.

Dobson and Kalish (1993) extended Green and Krieger's (1985) model to consider jointly both price setting and selection of a fixed number of products. In addition, they considered fixed costs per product. Dobson and Kalish (1993) sought to maximize the welfare or profit under the MSCR. van Ryzin and Mahajan (1999) integrated the product line purchasing and inventory management decisions. They used the homogeneous expected utilities of consumers to describe how consumers substitute for a favorite product not carried by a retailer; that is, they explained *selection-based substitution*. Li (2007) also studied the joint product line selection and purchasing decisions problem for a single period with different cost parameter values among products. Maddah and Bish (2007) considered joint decisions of selection, pricing, and inventory in the product line problem under the selection-based substitution. Gajjar and Adil (2010) and Irion et al. (2012) considered product line selection and pricing decisions using a mixed integer programming model that incorporated the space elasticity, which is based on the assumption that sales depends on product placement on the retailer's shelf.

Consumer substitution of product types, as introduced herein, is independent of the product inventory levels and reflects consumer preferences on product types and their combinations. We call this *static substitution of consumers*. Popular models used to analyze this substitution include the multinomial logit model (MNL). However, sometimes consumers also substitute goods when the retailer carries the favorite product, but it is out of stock; this type of consumer behavior is called *stockout-based substitution* or *dynamic substitution*, and in the decisions problem the consumer's choice represents the selection made from the available products in the product line of the retailer. Mahajan and van Ryzin (2001) considered dynamic substitution in which consumers take the available inventory into account before choosing product types to purchase. Burkart et al. (2012) and Mayer, Klein, and Seiermann (2013) also addressed the product line pricing problem under the dynamic substitution.

In dynamic substitution, the shelf space constraint of the retailer is crucial in product line decisions. Yücel, Karaesmen, Salman, and Türkay (2009) considered the shelf space constraint in their product line decisions of product selection, purchasing, supplier selection, and inventory management. However, their model did not incorporate consumer's preference.

In this paper, we present a joint product line selection, purchasing, and pricing decisions problem. The work is based on consumer choice and substitution in the retailer's product line and explains ways to make product line selecting, purchasing, and pricing decisions that maximize the retailer's total profit when the cost parameters across products differ. In addition, the dynamic substitution from the limited availability of shelf space and product inventory is addressed in the model.

3. Model

The proposed model features a retailer who chooses the types of products included in a product line as based on available shelf space in one selling season. In addition, the retailer sets the price and order quantity of each product type in a season in which no further replenishment is possible. When ordering a type of a product, the retailer incurs a fixed ordering cost that accounts for logistics and other administrative costs. The retailer considers the shelf space and restricts the total number of products ordered.

In the selling season, each consumer who arrives at the retailer's store has placed a preconceived, personal valuation on each type of product. Each consumer will choose the product that embodies the maximum difference between the valuations placed on it and the retail price such that the customer expects the value attributed to the product to exceed the retailer's price of it. For simplicity, the model limits each buyer to one unit of the product in the product line. We define the duration of time from the arrival of a consumer to the next consumer's arrival to the retailer by a *period* in our model. Each time a consumer buys a product, the retailer recognizes a holding cost for each unit of remaining product; this cost includes any deterioration value of a product or any interest cost incurred during the period. We assume that each product type may have a different holding cost rate.

Based on the assumptions that the retailer sets the prices of products from a set of several available prices and that the order quantity is an integer, an integer programming (IP) model was created to describe the retailer's pricing and ordering decisions. In addition, the choice of product types included in a product line is also described with an indicator variable.

To describe the problem, a sequence of events is delineated. At the beginning of a selling season, a retailer obtains the list of available products from several suppliers, who may offer different brands that satisfy high- or low-level market segments. Then, the retailer selects the products to order, the quantity of each selected product, and the prices to charge consumers to maximize the total profits during the selling period. In this decision-making process, the retailer must also consider consumer choice and relevant costs for purchasing and maintaining inventory after ordering.

A set of potential products is denoted by j $(j = 1,\ldots,J)$. The products have an inventory cost denoted by h_j, purchasing cost denoted by c_j, and ordering cost denoted by o_j. A time period i is denoted by t_i, which represents the interarrival time between consumer $i-1$ and i, with $t_0 := 0$. Also, we normalize the length of the selling season to 1 without loss of generality, which is imposed by $\sum_{i=1}^{I} t_i = 1$. Then, for product type j, we assume that a holding cost of h_j is incurred if one unit of that product type remains throughout the selling season. Based on this definition, the holding cost of product type j for time period i is calculated by $t_i h_j$ for each unit of product type j that is available for time period i.

A retailer chooses the products to offer on the shelf from a list of products available to purchase from suppliers. The retailer then determines the order quantity of product j, denoted by Q_j for $j = 1, \ldots, J$. Due to marketing considerations, the retailer chooses price P_j for each product j from a set of pre-defined discrete price points with N elements; $P_j \in \{P_{j1}, \ldots, P_{jn}\}$, where n indicates the price index such that $n = 1, \ldots, N$. There are I consumers who come to the retailer's store; they are numbered in one period indexed by i $(i = 1, \ldots, I)$. The inventory of product j available for the newly arrived consumer i is denoted by I_{ij}. If consumer i chooses product j, then the inventory of the product is reduced by1. Hence, $Q_j = I_{0j}$ and product j is available for consumer i if $I_{i-1,j} > 0$. With this definition of inventory of product j for consumer i (i.e., $I_{i,j}$), we can calculate the holding cost for period i by $t_i \sum_{j=1}^{J} I_{ij} h_j$.

The valuation the consumer places on product j (i.e., the reservation price) is denoted by V_{ij}. To make our problem computationally tractable to derive insights, we optimize the price of each product based on a given stream of consumers' valuations (V_{ij}) in the same manner as Burkart et al. (2012) and Mayer et al. (2013). Each consumer selects no more than one available product type, denoted by C_i, that provides the highest positive consumer surplus $(V_{ij} - P_{jn})$ as per the MSCR:

$$C_i = \arg max_{j\in\{1,\ldots,J\}} \left\{ V_{ij} - P_{jn} \middle| \left(I_{i-1,j} \ > \ 0 \right) \cap \left(V_{ij} - P_{jn} \geq 0 \right) \right\} \quad (1)$$

In Eq. (1), $I_{i-1,j} \ > \ 0$ means that product j is available for consumer i, and $V_{ij} - P_{jn} \geq 0$ means that consumer i's valuation of product j is higher than the price of product j. The maximum operator over j means that consumer i will buy the available product to obtain the maximum surplus. If $V_{ij} - P_{jn} \geq 0$ is not true for any available product, then consumer i leaves without buying anything. Under dynamic substitution, the available inventory level of a product may be successively reduced in each period. As a consequence of this depleted inventory, late-arriving consumers tend to face a restricted choice in products available for purchase.

The following indexes are used in the IP model for product line selection, pricing, and purchasing decisions of the retailer:

$i = 1, 2, \ldots, I$	Indices of the consumers
$j, k = 1, 2, \ldots, J$	Indices of the products
$n, m = 1, 2, \ldots, N$	Indices of the price points

While we set N as the number of price points for all product types, this model can address a situation in which each product type has varying price points. For example, if Product Type 1 has three price points and Product Type 2 has four price points, we set $N = \max\{3,4\} = 4$. Then, we set $P_{14}\ = 0$ or to infinity to exclude this choice from our model.

The parameters used in the model follow:

P_{jn}	nth price point of product j, i.e., $P_j \in \{P_{j1}\ , \ldots, P_{jn}\}$
S_{ijn}	Modified consumer surplus of consumer i for product j at price point P_{jn}
S_{ijn}	$= 1 + V_{ij} - P_{jn}$, if $V_{ij} - P_{jn} \geq 0$, or $S_{ijn} = 0$, otherwise
c_j	Unit purchasing cost of product j
h_j	Unit inventory cost of product j
o_j	Ordering cost of product j
V_{ij}	Reservation price of consumer i for product j
$type\_num$	Number of product types
S	Total shelf space
s_j	Occupied space for product j
t_i	Interarrival time of consumer i

Let the product that occupies the smallest shelf space be product k. Then, s_j is such that $s_k = 1$, which satisfies $s_j \geq 1$ for all j. This ensures that S has a greater value than the sum of orders for all products $(\sum_{j=1}^{J} Q_j \leq S)$. Then, S can be used as a proxy for the total order quantity, which is a useful way to maintain linearity in the model.

The decision variables used in this model follow:

x_{ijn}	1 : If consumer i chooses product j at the price point P_{jn} 0 : otherwise
g_{jn}	1 : If product j is offered at price point P_{jn} 0 : otherwise
τ_{ij}	1 : If product j is available for consumer i 0 : otherwise
y_j	1 : If product j is selected by the retailer 0 : otherwise
I_{ij}	Remaining inventory of product j after consumer i makes purchase
Q_j	Order quantity for product j

The objective function in Eq. (2) is used to maximize the retailer's total profit. It includes the total revenue (TR) as shown in Eq. (3) and the subtraction of the inventory cost (TIC) as shown in Eq. (4), total purchasing cost (TPC) as shown in Eq. (5), and total ordering cost (TOC) as shown in Eq. (6).

$$\text{Max}\, TP\,(g,\ y,\ Q) \ = \ \text{TR} - \text{TIC} - \text{TPC} - \text{TOC} \quad (2)$$

$$\text{TR} = \text{total revenue} = \Sigma_{i=1}^{I} \Sigma_{j=1}^{J} \Sigma_{n=1}^{N} x_{ijn} P_{jn} \quad (3)$$

$$\text{TIC} \ = \ \text{total inventory cost} \ = \sum_{i=1}^{I} \left(t_i \sum_{j=1}^{J} I_{ij} h_j \right) \quad (4)$$

$$\text{TPC} \ = \ \text{total purchasing cost} \ = \sum_{j=1}^{J} c_j Q_j \quad (5)$$

$$\text{TOC} \ = \ \text{total ordering cost} \ = \sum_{j=1}^{J} o_j y_j \quad (6)$$

The constraints for the retailer's decision problem are as follows:

$$\sum_{n=1}^{N} g_{jn} = 1 \quad \text{for all } j \quad (7)$$

$$\sum_{j=1}^{J} \sum_{n=1}^{N} x_{ijn} \leq 1 \quad \text{for all } i \quad (8)$$

$$I_{ij} = I_{i-1,j} - \sum_{n=1}^{N} x_{ijn} \quad \text{for all } \ i,\ j \quad (9)$$

$$\sum_{j=1}^{J} s_j Q_j \leq S \quad (10)$$

$$y_j \times S \geq \sum_{i=1}^{I}\sum_{n=1}^{N} x_{ijn} \text{ for all } j \tag{11}$$

$$\sum_{j=1}^{J} y_j \leq type\_num \tag{12}$$

$$\tau_{ij} \geq \sum_{n=1}^{N} x_{ijn} \text{ for all } i, j \tag{13}$$

$$\tau_{ij} \times S \geq I_{i-1,j} \text{ for all } i, j \tag{14}$$

$$\tau_{ij} \leq I_{i-1,j} \text{ for all } i, j \tag{15}$$

$$x_{ijn} \leq S_{ijn} \times g_{jn} \text{ for all } i, j, n \tag{16}$$

$$S_{ijn}\left(\tau_{ij} + g_{jn} - 1\right) \leq \sum\nolimits_{k=1}^{J} \sum\nolimits_{m=1}^{N} S_{ikm}x_{ikm} \quad \text{for all } i, j, n \tag{17}$$

$$\sum\nolimits_{i=1}^{I} t_i = 1,\ t_0 = 0 \tag{18}$$

$$Q_j, I_{ij} \geq 0 \text{ for all } i, j \tag{19}$$

$$Q_j = I_{0j} \text{ for all } j \tag{20}$$

$$x_{ijn}, \tau_{ij}, g_{jn}, y_j \in \{0, 1\} \text{ for all } i, j, n \tag{21}$$

Constraint (7) ensures that for each product exactly one price point can be chosen. Constraint (8) means that every consumer can purchase, at most, one product or leave without purchasing anything. Constraint (9) illustrates that the inventory level of a product is reduced by 1 after a consumer makes a purchase. Constraint (10) accounts for limited shelf space for a product line such that the total occupied space with total order quantity of products included in the product line cannot exceed the shelf space of the retailer. Constraint (11) suggests that consumers can only buy the products that a retailer has selected to sell and ordered at the beginning of the period. Constraint (12) suggests that the number of selected types of items cannot exceed the retailer's limit for the product line. Constraint (13) indicates that the purchase behavior of each consumer toward one product can be realized if and only if the product is available. Constraint (14) forces τ_{ij} to become 1 if product j is available for consumer i. Constraint (15) forces τ_{ij} to become 0 if product j is unavailable to consumer i. Constraint (16) represents the purchase of a product at a selected price point, which is only possible if and only if the corresponding consumer surplus is non-negative. Constraint (17) forces a consumer into purchasing when at least one consumer surplus for any product is non-negative. At the same time, the purchased product should be available at a certain price point when the consumer arrives. The constraint also ensures that consumers buy the product that yields the highest possible consumer surplus. Constraint (18) implies that for the duration of the problem the total inter-arrival times is normalized to 1 with the initial period set to 0. Constraint (19) ensures that the decision variables are nonnegative and Constraint (20) initializes the inventory level for each product. Constraint (21) demonstrates the binary nature of the decision variables.

4. The hybrid genetic algorithm

A genetic algorithm (GA) is a population-based search and optimization method inspired by Darwin's theory of evolution. The two main concepts of evolution, natural selection and genetic dynamics, play key roles in this method. A GA and basic principles were first introduced by Holland (1975) and have been widely used in various areas during the last four decades. The GA has the advantage of being usable with other techniques to improve search performance. An algorithm incorporating a search method within a GA is called a "hybrid genetic algorithm" (HGA).

While some heuristics and generic algorithms (GAs) have been considered, we see that the existing heuristics and GAs cannot be used for the joint product line decisions problem. Due to the complexity in our formulation including holding cost, pricing, and ordering for a large scale of problems that has not been considered together, we need to develop a new HGA in this paper. We compare our paper and other related papers with similar heuristics and algorithms in Table 1.

A hybrid generic algorithm (HGA) is proposed to solve large scale problems in this study. The suggested HGA structure is shown in Fig. 1. As represented in the figure, after the HGA is initiated over the population and the generation counter is set to 1, chromosomes are generated and selected before crossover and mutation are conducted. After chromosome manipulations, the fitness of the solution is evaluated and the algorithm repeats until the termination condition is satisfied.

4.1. Chromosome generation

In a GA, the solution must be represented in proper form. Traditionally, chromosomes are represented as simple binary strings; however, this representation is not useful for solving the joint product line decisions problem. Therefore, a chromosome consisting of three parts, one for each decision in the problem, is explained. One part represents the product selection for a product line (selection), the other shows the prices for each selected product (pricing), and a third stands for the order quantity of each selected product (purchasing). An example of a chromosome in this study is presented in Table 2.

The efficiency of the algorithm greatly depends on the initial solution. Therefore, a simple heuristic is proposed to create a fea-

Table 1
Comparisons of related papers.

Author (year)	Consideration	Methodology	Limitation
Kraus and Yano (2003)	Selection, pricing, consumers probabilistic choice	MIP heuristics	No purchasing decision considering variable costs No consideration on Dynamic Substitution Only small and medium size problems up to 2000 consumers
Aydin and Porteus (2008)	Pricing, purchasing, price-based substitution, demand function	MIP heuristics	No selection decision Only small size problem up to 1000 consumers
Yücel et al. (2009)	Selection, purchasing, demand substitution, supplier selection	IP	No pricing decision Only small size problem up to 100 but provide some managerial insights
Burkart et al. (2012)	Pricing, consumers max-surplus choice, dynamic substitution	MIP heuristics	No selection and purchasing decisions Large size of price points but only small size of consumers up to 1000
This paper	Selection, pricing, purchasing, consumer max-surplus choice, dynamic substitution	IP hybrid genetic algorithm	

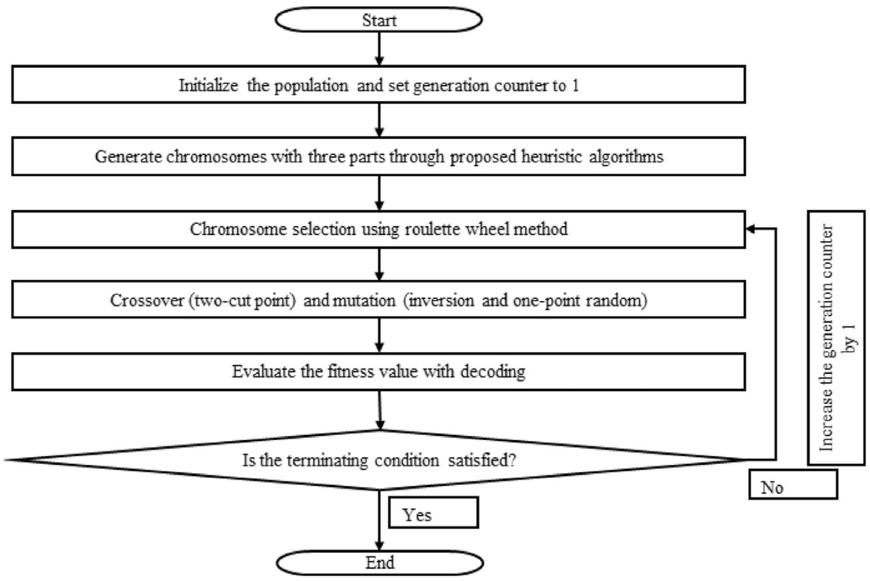

Start
Initialize the population and set generation counter to 1
Generate chromosomes with three parts through proposed heuristic algorithms
Chromosome selection using roulette wheel method
Crossover (two-cut point) and mutation (inversion and one-point random)
Evaluate the fitness value with decoding
Is the terminating condition satisfied?
Increase the generation counter by 1
No
Yes
End

Fig. 1. Flowchart of the proposed hybrid genetic algorithm.

Table 2
Example of a chromosome.

Product	1	2	3	4	5	6	7	8	9	10
Selection	1	0	0	1	1	1	0	1	0	0
Pricing	30	35	45	60	65	75	80	85	95	95
Purchasing	84	65	70	100	41	116	98	50	62	88

sible initial solution. This heuristic uses a sorting procedure based on the consumer choice process. The detailed steps for the heuristic are as follows:

Step 1: Get the defined beta-distribution reservation prices for each potential product, each of which features parameters α_j and β_j, and regard them as reservation prices V_{ij} for $i = 1$.

Step 2: Calculate the mean of each reservation price $E(V_{ij})$:

$$E\left(V_{ij}\right) = \frac{\alpha_j}{\alpha_j + \beta_j} \times 100.$$

Step 3: Calculate the critical consumer surplus cv_j for each product:

$$cv_j = E\left(V_{ij}\right) - c_j - h_j$$

Step 4: Order one of the products such that the maximum critical consumer surplus value is reached and create the critical consumer surplus table.

Step 5: Update the table by deleting the selected product.

Step 6: Go back to Step 4 until *type_num* products are assigned.

4.2. Chromosome selection

In an HGA, the generic operator consists of selection, crossover, and mutation. The selection is the stage in which individual chromosomes are chosen from a population. In this algorithm, parents are chosen through the roulette wheel method. In this approach, the fitness of each chromosome is calculated and used to associate a probability of selection with each individual chromosome. Usually a roulette wheel is divided into proportions such that when it is rotated, a ball lands in one of the possible partitions. Likewise, with this method, a solution is selected from a range of possibilities such that there is a chance that some weaker solutions may survive through the selection process. This inclusion of the weaker solutions proves advantageous as it may contain some components useful in the recombination process.

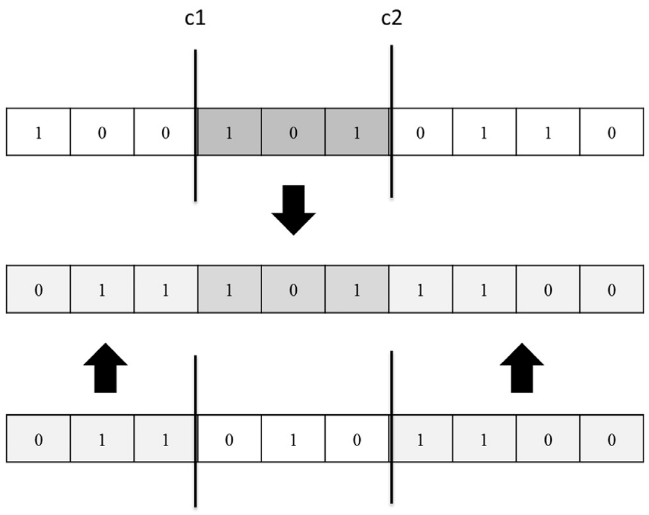


Fig. 2. Two-cut point crossover of genes for product selection.

4.3. Chromosome crossover and mutation

The crossover operation is a diversification mechanism that enables the GA to generate a solution in previously unvisited areas of the gene pool. In this algorithm, a two-cut crossover operator is used. Crossover occurs with a probability of P_c.

To conduct a crossover, two arbitrary cutting lines, $c1$ and $c2$, are placed on P_1. The genes between two cutting lines are copied to the same location on O_1. The rest of the genes on O_1 are copied in order of sequence from the genes on P_2. O_2 is built through the same process.

An example of crossover is illustrated in Fig. 2. After crossover of the product selection, prices and order quantities will be adjusted correspondingly. The processes for product and price selections are illustrated in Figs. 2 and 3.

In a GA, mutation produces a random change in a chromosome. The main difference between mutation and crossover is that the mutation operators affect one chromosome; that is, they are unary, while crossovers are binary operators. Mutation plays the crucial role of replacing genes of chromosomes during evolution

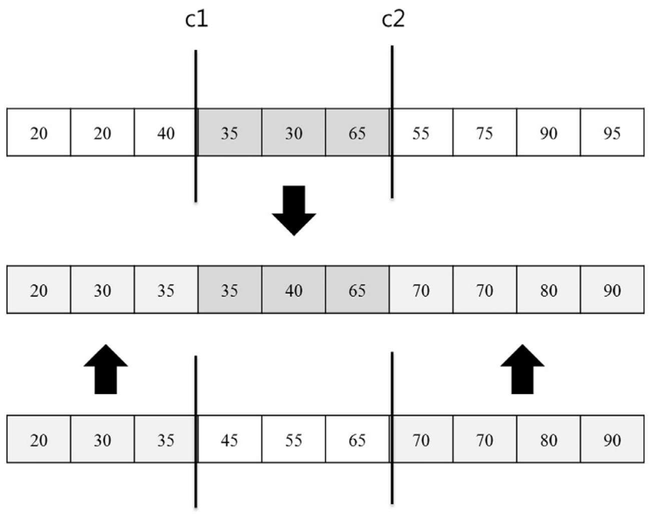


Fig. 3. Two-cut point crossover of genes for price assignment.

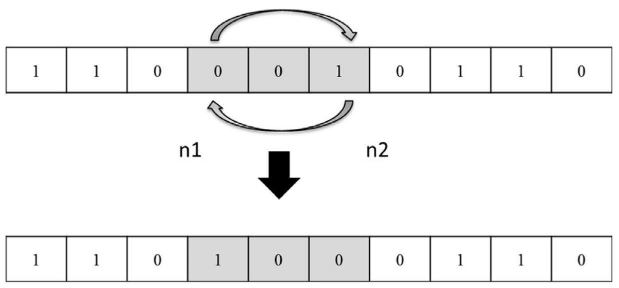


Fig. 4. Inversion mutation of a gene for product selection.

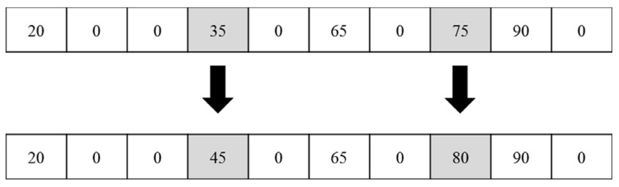


Fig. 5. Random point mutation of a gene for price assignment.

so that they can maintain diversity in the population. Genes carry the characteristics that make each individual unique.

In this study, mutation occurs with a probability of P_m. For product selection, the inversion mutation is used. The combination of $n1$ and $n2$ cannot contain more genes than exist in the chromosome. Furthermore, $n1$ and $n2$ should not consist of the same number. The gene with the smaller value is designated as $n1$. The genes from $n1$ to $n2$ are rearranged in reversed order; see Fig. 4. For price assortment and order quantity assignment, a random point mutation operator is used. Fig. 5 shows an example of random point mutation for price assignment.

4.4. Improved genetic operators

The operators created to vary the gene pool are widely used for GAs. However, due to the complexity of the joint product line decisions problem, the chromosome required for the model contains three parts, which may result in a badly converged GA. This possibility is shown through some simple examples. Furthermore, to increase the diversity in the search space and avoid premature convergence of the GA, two more operators were added to the model: dynamic mutation probability and population catastrophe.

4.4.1. Dynamic mutation probability

The dynamic mutation probability approach added to this study was first proposed by Deif et al. (2012). The mutation is conducted by dividing the chromosomal population into two groups and assigning different mutation operators to each one of them. For chromosomes with fitness values higher than average, a high value P_m is set to increase the diversity of the search space and to ensure the solution proceeds all the way to an optimal point. For low fitness chromosomes, P_m is set to a low value to reduce the computation time.

The algorithm can be explained as follows:

In a given population, let f_c be the fitness value of any chromosome c and let f_{avg} be the average fitness value. Then, the mutation probability P_m is determined as follows:

If $f_c \geq f_{avg}$, a high mutation probability is assigned.

If $f_c < f_{avg}$, a low mutation probability is assigned.

4.4.2. Population catastrophe

The genetic catastrophic algorithm (GCA) was first presented by Jin and Li (1997). It is used to recover the population diversity when premature convergence has occurred.

Generally, the catastrophic operation includes two steps:

(1) Remove a certain number of individuals with low fitness values from the current population.
(2) Regenerate individuals via various methods and put them into the current population to replace the removed ones.

Through the addition of various new individuals to the current population, the diversity of the population can be greatly improved.

4.5. Fitness function and termination conditions

The fitness function plays a role similar to that of the environment in evolution. Each individual in the population represents a potential solution to the problem. A fitness function is computed for each string in the population and the string with the maximum fitness function value is selected as a possible solution to the problem. Eq. (2) is used as a fitness function in the proposed GA. To calculate the fitness value, information on all the individual and product-specific reservation prices is necessary. Hence, 30 scenarios were developed for each class and the average was calculated in each scenario. These calculations were made with the IBM ILOG CPLEX Optimization Studio.

The populations were listed in descending order of fitness value and only the top 50% were included in the new population. The operation is terminated in one of two ways. When the fitness value reached a fixed number of generations, iteration of the genetic operation is completed and the best chromosome is given as a solution. When the fitness value of the best chromosome has not been improved over a fixed number of generations, the operation will be stopped.

5. Computational experiments

The IP model in this study was conducted using the IBM ILOG CPLEX Optimization Studio for the exact solution. The IP model with the HGA was run in C# language in Windows 7 on a PC with Intel Core 2.0 gigahertz. Although the computer specifications were appropriate for the job, these models were unable to solve problems as large as those with 2000 consumers, 5 product types, and 10 price points within the pre-set time limit. However, HGAs can find solutions within a reasonable computation time for large scale problems.

Table 3
Class components.

Class components	Data sets	Explanations
C_1	{3, 5, 8, 10}	Number of product types ($type\_num \in C_1$)
C_2	{300, 500, 800, 1000}	Number of consumers ($I \in C_2$)
C_3	{3,5,10}	Number of pre-defined price points ($N \in C_3$)
C_4	{low, high}	Overlap D of the density functions of the reservation prices ($D \in C_4$)

Table 4
Price points.

Class $\mathbf{C}_3$ ($\boldsymbol{N} \in \mathbf{C}_3$)	Data sets
$N = 3$	{33, 67, 100}
$N = 5$	{20, 40, 60, 80, 100}
$N = 10$	{10, 20, 30, 40, 50, 60, 70, 80, 90, 100}

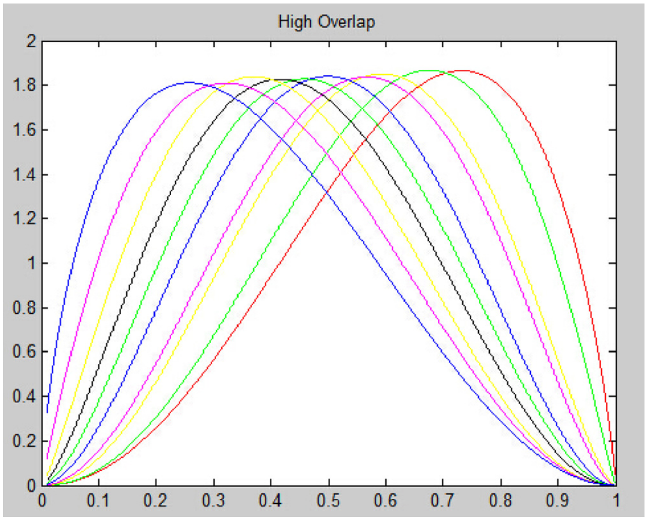


Fig. 6. High overlap of the probability density function (PDF).

5.1. Design of experiment

To provide a large variety of real world instances for the joint product line decisions problem, $\prod_{i=1}^{4} C_i = 96$ problem classes were constructed with four different sets of components $C_1, \ldots, C_4$. The details for each class of components are illustrated in Table 3.

Due to marketing considerations, a retailer must choose a price P_j for each product j out of a set of pre-defined discrete price points: P_{jn} $(n = 1, \ldots, N)$; i.e., $P_j \in \{P_{j1}, \ldots, P_{jn}\}$ in Table 4. For simplicity and tractability, we assumed that price points across all products were identical. Also, we set the interarrival times identical for all products (i.e., $t_i = 1/I$); however, specific peak and non-peak times of consumer arrival in the retail store can be easily incorporated in the calculation of inventory holding cost (Eq. (4)) by adjusting the time period (i.e., t_i for $i = 1,\ldots,I$) such that the peak times are relatively short compared to non-peak times.

We set consumer reservation price V_{ij} for product j by consumer i. Just as Burkart et al. (2012), we used Beta (α_j, β_j) distribution to collect data on consumer reservation prices. With the beta distribution, various types of consumers' reservation prices across different products can be incorporated.

First, the reservation prices across products are relatively similar (i.e., a high-overlap class). The behaviors of consumers in this class are associated with low price elasticity values and relatively low maximum values of the density function, which means that they do not have differential preferences for different types of products. Fig. 6 shows the probability density function (PDF) of consumers' reservation prices for the high-overlap class. Second, consumers may show explicit preferences for specific products. This occurs when reservation prices does not overlap across each product (i.e., a low-overlap class). Fig. 7 presents the PDF of reservation prices for the low-overlap case.

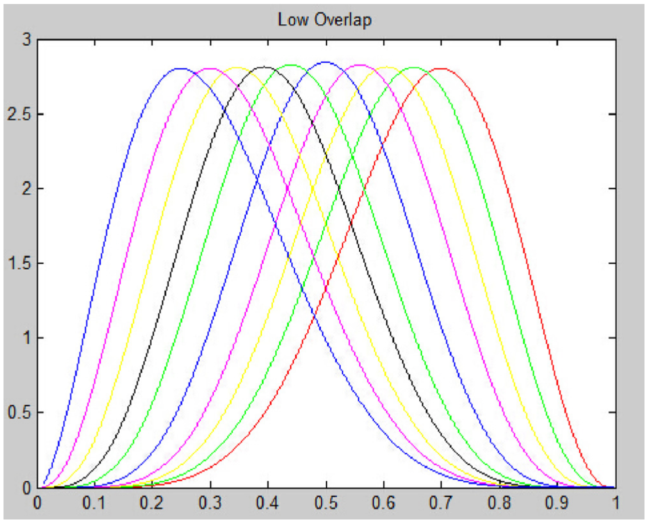


Fig. 7. Low overlap of the probability density function (PDF).

In our numerical experiment, consumer reservation prices are normalized with the beta distribution and then scaled to the interval [0, 100]. The overlap $D \in C_4$ of the density functions of reservation prices are reflected in the data. The details for the consumer reservation prices sets are shown in Table 5.

The values of other parameters that we used for the numerical experiment are presented in Table 6.

5.2. Results and insights

In this section, the results of experiments for 96 classes conducted with the devised mathematical model are reported. These experiments were used to maximize the total profit for a retailer who could select between 10 potential products. For each problem class, 30 scenarios were generated by random reservation prices drawn for each consumer and each product; then, average results of these scenarios were calculated using CPLEX. Computation times to obtain the exact solutions with CPLEX are summarized in Table 7. In the table, "X" indicates that the solution was unobtainable within a reasonable period of time. The HGA always yields a solution, and computation times with the HGA are shown in Table 8. To validate the proposed algorithms, the optimal solutions were compared with solutions from the HGA. In Table 9, we calculate the gap of the retailer's profits obtained from CPLEX and the HGA. The table shows that the gap remains a reasonably small (3%) in most of our numerical experiments.

Some managerial insights can be confirmed through the results of these experiments. First, the relationship between a retailer's profit and the *type_num* limit indicates that a large *type_num* value does not always yield a high profit for the retailer; an optimal number of product types applies to a product line. Fig. 8 illustrates this relationship by presenting the results of classes featuring low overlap and 3 pre-defined price points. Out of 10 products, the results indicate that for the $I = 300$, 500, and 800 classes, the optimal *type_num* remains 8, although total 10 number of the products is available (i.e., type_num $\leq$ 10).

Second, for a small *type_num* value, the profit of the retailer for reservation prices from the high overlap PDF is relatively similar to the profit from the low overlap PDF. However, for large values

Table 5
Reservation prices.

Class $C_4(D \in C_4)$	**Parameters**	
	α_j	β_j
$D=$ high overlap	{1.7, 2.0, 2.3, 2.5, 2.7, 2.9, 3.1, 3.2, 3.3, 3.2}	{3.0, 3.1, 3.2, 3.1, 3.0, 2.9, 2.6, 2.5, 2.1, 1.8}
$D=$ low overlap	{3.0, 3.7, 4.4, 5.1, 5.8, 6.6, 7.1, 7.3, 7.4, 7.3}	{7.0, 7.3, 7.4, 7.3, 7.1, 6.6, 5.8, 5.1, 4.0, 3.7}

Table 6
Other parameter values.

Parameters	Values
Unit inventory costs h_j, $j=1,\ldots,10$	{0.5, 1.0, 1.2, 0.7, 1.5, 0.9, 2.0, 1.5, 1,0, 0.8}
Unit purchasing cost c_j, $j=1,\ldots,10$	{1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0}
Ordering costs o_j, $j=1,\ldots,10$	{50, 40, 125, 100, 160, 255, 200, 175, 295, 75}
Occupied space l_j, $j=1,\ldots,10$	{1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0}

Table 7
Computation times to obtain the exact solution using CPLEX.

		$I=300$			$I=500$			$I=800$			$I=1000$		
	Overlap	$N=3$	$N=5$	$N=10$	$N=3$	$N=5$	$N=10$	$N=3$	$N=5$	$N=10$	$N=3$	$N=5$	$N=10$
3	High	100	156	739	318	563	2617	789	1657	7449	1076	3292	20,979
	Low	178	228	469	382	594	1263	865	2124	6887	X	2854	8518
5	High	523	1568	9562	1842	4133	X	3154	X	X	X	X	X
	Low	135	919	2154	403	2924	X	1072	11,303	X	4941	X	X
8	High	1253	1872	14,650	3332	6810	X	12,803	17,155	X	X	X	X
	Low	67	257	8595	185	1173	22,476	336	5249	X	620	6232	X
10	High	1034	951	3855	4575	7289	X	8919	22,652	X	18,661	X	X
	Low	55	265	8375	131	885	20,087	252	1657	X	272	5907	X

X indicates that CPLEX does not produce a solution for 24,000 seconds.

Table 8
Computation times to obtain the heuristic solution with the HGA.

		$I=300$			$I=500$			$I=800$			$I=1000$		
	Overlap	$N=3$	$N=5$	$N=10$	$N=3$	$N=5$	$N=10$	$N=3$	$N=5$	$N=10$	$N=3$	$N=5$	$N=10$
3	High	5953	5352	5183	9117	8100	7982	13,821	12,259	12,095	17,757	16,686	15,994
	Low	5520	4971	4807	9257	8305	8203	14,465	12,850	12,453	18,246	17,793	16,935
5	High	4337	4028	3997	6554	6110	6056	10,935	10,262	10,080	13,778	13,019	12,843
	Low	4054	3837	3772	6582	6229	6157	10,978	10,442	10,333	14,152	13,959	13,233
8	High	3271	3441	3308	4998	5269	5077	8176	8007	7796	10,542	10,517	10,195
	Low	3339	3171	3083	5223	5193	5076	8217	8172	8195	11,576	11,429	11,107
10	High	2786	2450	2268	4274	3795	3571	6479	6053	5913	8650	8107	8059
	Low	2583	2279	2134	4540	4117	3722	6882	6241	5942	9397	9383	9390

Table 9
Gap between the exact solution from CPLEX and the HGA (%).

		$I=300$			$I=500$			$I=800$			$I=1000$		
	Overlap	$N=3$	$N=5$	$N=10$	$N=3$	$N=5$	$N=10$	$N=3$	$N=5$	$N=10$	$N=3$	$N=5$	$N=10$
3	High	3.89%	1.06%	2.80%	0.99%	8.62%	1.82%	0.31%	1.10%	0.10%	1.02%	0.92%	3.48%
	Low	0.98%	0.12%	5.69%	2.62%	1.29%	4.21%	0.60%	1.65%	1.99%	X	1.06%	5.18%
5	High	2.14%	3.36%	3.73%	1.60%	2.47%	X	0.30%	X	X	X	X	X
	Low	0.17%	5.25%	3.95%	0.93%	1.31%	X	2.22%	2.52%	X	3.59%	X	X
8	High	0.35%	2.54%	1.47%	0.58%	0.46%	X	0.53%	4.08%	X	X	X	X
	Low	3.81%	3.79%	1.94%	3.06%	2.29%	4.70%	2.34%	0.74%	X	2.25%	2.28%	X
10	High	0.63%	3.58%	4.86%	2.63%	0.71%	1.65%	1.41%	4.93%	X	4.94%	X	X
	Low	3.98%	4.21%	2.82%	2.43%	3.31%	5.09%	3.89%	2.45%	X	4.23%	4.72%	X

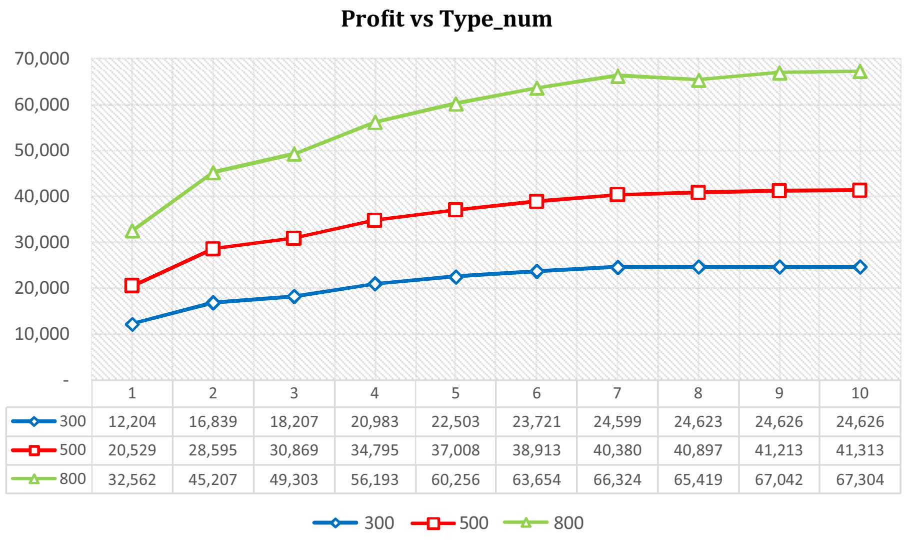
X indicates that CPLEX does not produce a solution for 24,000 seconds.

of *type_num*, the difference of the retailer's profits from the high overlap and low overlap PDFs is relatively large. Reflecting consumer characteristics, a large value for *type_num* tends to mean a relatively high retailer's profit for highly differentiated consumers (with low overlap) than is achieved when consumers are poorly differentiated (with high overlap). Fig. 9 illustrates this relationship with 300 consumers and 3 pre-defined price point classes.

Third, Fig. 10 shows that pre-defined price points are associated with higher profit. This finding is consistent with common knowledge.

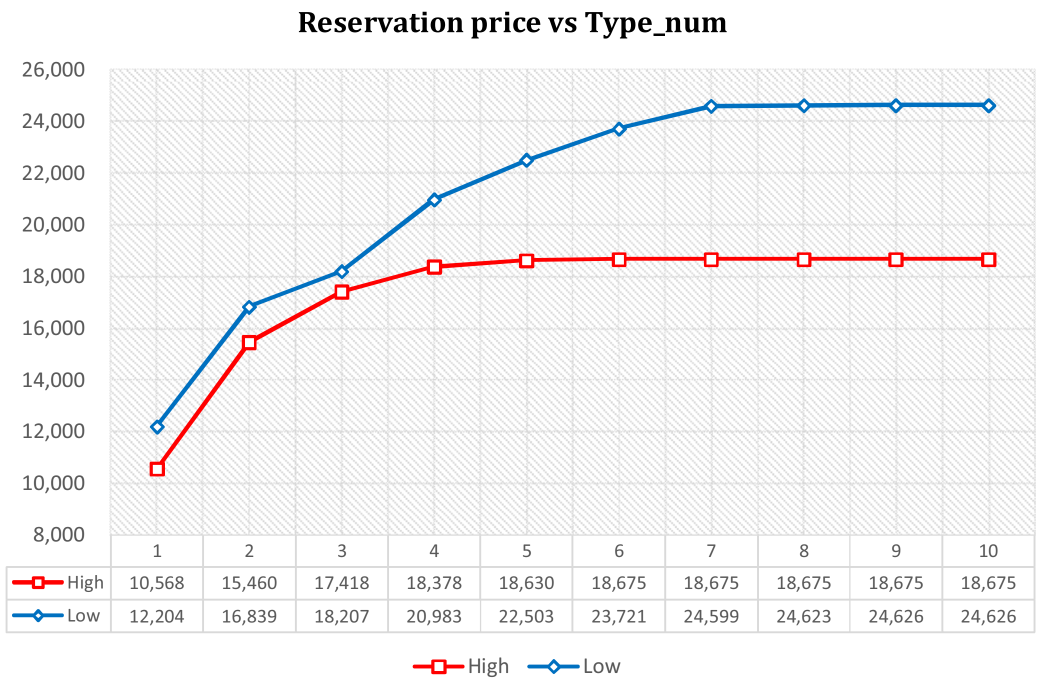
6. Conclusions

In this study, the joint product line decisions problem is used to determine optimum product selection, purchasing, and pricing for a retailer. In making these decisions, the retailer incorporates consumer choice processes, ordering and holding costs for each product, and the resulting dynamic substitution in each period, which depends on product availability. An integer programming model was developed for this joint product line decisions problem. The model was designed to maximize a retailer's profit, which includes



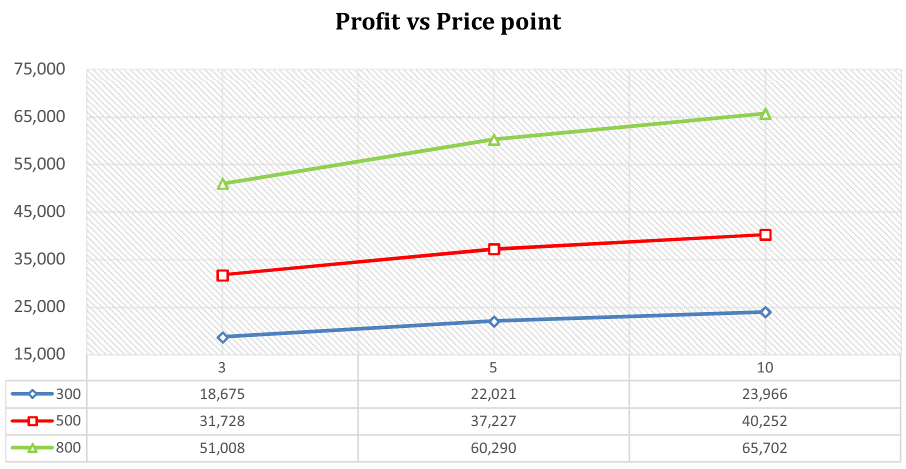
	1	2	3	4	5	6	7	8	9	10
300	12,204	16,839	18,207	20,983	22,503	23,721	24,599	24,623	24,626	24,626
500	20,529	28,595	30,869	34,795	37,008	38,913	40,380	40,897	41,213	41,313
800	32,562	45,207	49,303	56,193	60,256	63,654	66,324	65,419	67,042	67,304

Fig. 8. Results of low overlap and 3 pre-defined price points.



	1	2	3	4	5	6	7	8	9	10
High	10,568	15,460	17,418	18,378	18,630	18,675	18,675	18,675	18,675	18,675
Low	12,204	16,839	18,207	20,983	22,503	23,721	24,599	24,623	24,626	24,626

Fig. 9. Results of 300 consumers and 3 pre-defined price points.



	3	5	10
300	18,675	22,021	23,966
500	31,728	37,227	40,252
800	51,008	60,290	65,702

Fig. 10. Results of high overlap classes.

the total revenue after eliminating inventory, purchasing, and ordering costs. A large variety of real-world instances of the joint product line decisions problem was considered such that the computational experiments included 96 problem classes of four different sets of components.

To solve the model, an efficient HGA was developed and proposed. The HGA combines effective heuristic algorithms with the genetic algorithm by taking the efficiency of solution processes and the quality of solutions into account. We found that the proposed algorithm achieves a solution with reasonably small error compared to the optimal solution obtained by CPLEX for small problems. Furthermore, for large problems, we observe that the HGA overcame the computational burden under which CPLEX fails to solve the problem. Therefore, the HGA was shown to be a useful method to solve the joint decisions model of a product line for large scale problems.

Based on the results, some useful managerial insights were derived. First, an optimal number of product types characterizes the product line selection problem. Second, the impact of consumer valuation (reservation price) on the optimal number of product types in the product line was determined. Lastly, a common sense solution in selecting the number of price points was supported as a way to maximize the retailer's profit.

For further study, the demand formulation could replace the reservation prices. Also, because the model presented is based on the assumption of lost sales, a penalty cost for unsatisfied demand could be incorporated in a follow-up study and would likely reveal interesting findings.

Acknowledgments

The authors are grateful to valuable comments by the editor and referees to improve earlier versions of the manuscript. This research was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning [grant number 2015R1A2A1A15053948].

References

Aydin, G., & Porteus, E. (2008). Joint inventory and pricing decisions for a selection. *Operations Research, 56*, 1247–1255.

Burkart, W. R., Klein, R., & Mayer, S. (2012). Product line pricing for services with capacity constraints and dynamic substitution. *European Journal of Operational Research, 219*, 347–359.

Deif, S., Kamal, H. A., & Tawfik, M. (2012). Enhancing genetic algorithms using a dynamic mutation value approach: An application to the control of flexible Robot Systems. *International Journal of Artificial Intelligent Systems and Machine Learning, 4*(1), 9–16.

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.

Gajjar, H. K., & Adil, G. K. (2010). A piecewise linearization for retail shelf space allocation problem and a local search heuristic. *Annals of Operations Research, 179*, 149–167.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. University of Michigan Press.

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

Irion, J., Lu, J.-C., Al-Khayyal, F., & Tsao, Y.-C. (2012). A piecewise linearization framework for retail shelf space management models. *European Journal of Operational Research, 222*, 122–136.

Jin, X., & Li, Z. (1997). Genetic-catastrophic algorithms and its application in nonlinear control system. *Journal of System Simulation, 9*(2), 111–115.

Kraus, U. G., & Yano, C. A. (2003). Product line selection and pricing under a share-of-surplus choice model. *European Journal of Operational Research, 150*, 653–671.

Li, Z. (2007). A single-period assortment optimization model. *Production and Operations Management, 16*, 369–380.

Maddah, B., & Bish, E. K. (2007). Joint pricing, assortment, and inventory decisions for a retailer's product line. *Naval Research Logistics, 54*(3), 315–330.

Mahajan, S., & van Ryzin, G. (2001). Stocking retail assortments under dynamic consumer substitution. *Operations Research, 49*(3), 334–351.

Mayer, S., Klein, R., & Seiermann, S. (2013). A simulation-based approach to price optimization of the mixed bundling problem with capacity constraints. *International Journal of Production Economics, 145*(2), 584–598.

Mayer, S., & Steinhardt, C. (2016). Optimal product line pricing in the presence of budget-constrained consumers. *European Journal of Operational Research, 248*, 219–233.

Monroe, K. B., Sunder, S., Wells, W. A., & Zoltners, A. A. (1976). A multi-period integer programming approach to the product mix problems. In K. Bernhardt (Ed.), *Proceedings of the American marketing Association Meeting* (pp. 493–497).

Shugan, S. M., & Balachandran, V. (1977). *Working paper*. University of Rochester.

van Ryzin, G., & Mahajan, S. (1999). On the relationship between inventory costs and variety benefits in retail selections. *Management Science, 45*, 1496–1509.

Yücel, E., Karaesmen, F., Salman, F., & Türkay, M. (2009). Optimizing product selection under customer-driven demand substitution. *European Journal of Operational Research, 199*, 759–768.

Zufryden, F. S. (1982). Product line optimization by integer programming. *Presented at the Spring 1982 ORSA/TI MS Conference*.