Computers & Operations Research 37 (2010) 1797–1804

Contents lists available at ScienceDirect

Computers & Operations Research

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

Metaheuristics for assortment problems with multiple quality levels

Mark H. McElreath, Maria E. Mayorga *, Mary E. Kurz

Department of Industrial Engineering, Clemson University, 110 Freeman Hall, Clemson, SC 29634, USA

ARTICLE INFO

Available online 1 February 2010

Keywords:
Assortment planning
Metaheuristics

ABSTRACT

The assortment planning problem involves choosing an optimal product line, as defined by a set of products with specific attributes, to offer consumers. Under a locational choice model in which products are differentiated both horizontally (by variety attributes) and vertically (by quality attributes), an optimal assortment, whose attributes have only been partially characterized, may consist of multiple quality levels. Using previous analytical results, we approximate the optimal assortment for make-to-order and static substitution environments. We test the appropriateness and compare the performance of three metaheuristic methods. These metaheuristics can easily be modified to accommodate different consumer preference distribution assumptions.

1. Introduction

Assortment planning problems can roughly be divided into two groups, depending on how consumer choice is modeled—discrete versus locational choice. The appropriate model depends on the problem settings. (Mayorga presents a detailed review of existing literature that focuses on assortment decisions (Chapter 3, [7]).) In this paper, we focus on locational choice models under two different retail environments. In a make-to-order environment, consumers are willing to wait for their most preferred product to be delivered. In a make-to-stock environment with static substitution, consumers will not purchase if their most preferred product is not in stock. Thus inventory decisions must also be made in the static substitution environment.

A locational model for consumer choice was first proposed by Lancaster [6]. Under a locational choice model, the location of a product specifies its set of attributes; similarly each individual is characterized by her ideal point in the attribute space. The utility a consumer gets from a product is a function of the distance between the product's location and her ideal point. Thus, the locational choice model is a utility based model where the firm can control the rate of substitution between products by choosing their locations relative to each other.

Gaur and Honhon [2] are the first to study the assortment problem under a locational choice model while incorporating inventory decisions; they solve the static substitution problem, using a locational choice model with horizontal differentiation. Under horizontal differentiation, consumer preferences for an attribute can be modeled as a location on a line (e.g. shoe size) and the distribution of these preferences is known. They show that, under static substitution, the optimal assortment consists of products distributed on the attribute space such that there is no substitution among them and market coverage is continuous. When the distribution of consumer preferences for the horizontal attribute is uniform, the optimal assortment (location of products on the variety space and number of products) is known. On the other hand, when the distribution of consumer preferences is unimodal, the optimal solution can be found by conducting a single variable line search. Thus, we see that when consumer choice is based only on horizontal differentiation, the optimal assortments can be easily found.

Unfortunately, computational tractability is lost when vertical differentiation among products is allowed. Mayorga [7–9] considers an assortment planning problem in which products are both horizontally and vertically differentiated. With only horizontal differentiation, consumer choice is based only on an individual's idiosyncratic preferences (e.g., black vs. white earphones). The addition of vertical differentiation allows another type of consumer choice, based on consumers' taste for quality (e.g., regular vs. noise canceling earphones). The assortment and inventory management problem under different substitution environments is investigated by Mayorga [7,8], for the case that consumer preferences for the horizontal attribute are distributed uniformly, and extended by Mayorga et al. [9] to consider more general distribution assumptions. In the case of static substitution, the inventory and assortment problems completely decouple as was the case with Gaur and Honhon [2], such that given an assortment, the optimal inventory levels can be obtained analytically. The assortment problem, however, proves more difficult. Mayorga et al. [9] show that some properties of the optimal horizontal attributes are the same as in [2]; that is, products are spaced such that there is no substitution between

* Corresponding author. Tel.: +1 864 656 6919; fax: +1 864 656 0795.
E-mail address: mayorga@clemson.edu (M.E. Mayorga).

0305-0548/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2010.01.011

them and market coverage is continuous. However, to find the optimal assortment using these properties, one still needs to choose the number of products and the quality level of each product. In general, solving the problem to optimality requires an enumerative search over all possible assortment sizes and quality combinations. Mayorga et al. [9] show that the problem can be greatly simplified in some cases. In particular, they give conditions under which the optimal assortment is composed of products that all have the same quality level, reducing the problem to the previously solved problem with only horizontal differentiation. While they show that over a large region of the parameter space these conditions are indeed satisfied, we are left with an open problem when the conditions are not satisfied. In this case, the optimal assortment can be of mixed quality levels and neither the optimal number of products to carry or attributes are known.

In this paper we are interested in approximating the optimal assortment, even if it consists of multiple quality levels, for both the make-to-order and make-to-stock with static substitution environments. When the distribution of consumer preferences for the horizontal attribute is uniform, the problem can efficiently be solved to optimality [10]. The complexities introduced when consumer preferences follow a unimodal distribution are discussed and the assortment problem with multiple quality levels is formulated. We then provide an introduction to the metaheuristic techniques utilized and discuss the results of a computational study comparing the solution quality of these techniques. We show that these methods provide high quality solution even in cases where the optimal solution has not yet been characterized. To our knowledge, metaheuristic techniques have not been applied to solve assortment problems. The goal of this paper is to test the viability of using different metaheuristic techniques to solve assortment planning problems when the optimal solution cannot be found analytically. Thus, instead of focusing on creating the most efficient single metaheuristic that is well suited to solve this assortment problem, we test several metaheuristics which can be easily adapted under different assumptions. In later work, we investigate the use of a special purpose genetic algorithm to solve a more difficult problem (see Section 7).

The rest of the paper is organized as follows. Section 2 provides an overview of the assortment problem description and motivates the need for a metaheuristic approach. Results that will be useful in forming our metaheuristic approach are described in Section 3. Section 4 describes the metaheuristic techniques used. The design of the computational experiments and results are provided in Sections 5 and 6. Lastly, our conclusions are presented in Section 7.

2. Overview to the assortment problem

In this section we will describe the particulars of the assortment problem. We attempt to make the description as succinct as possible as these problems have been described by others, e.g. Gaur et al. [2], Mayorga [8,7], and Mayorga et al. [9]. While the definitions and terms used are conventionally found in the literature, the notation introduced is specific to this paper.

Whether the firm operates in a make-to-stock or static substitution environment it seeks to maximize expected profit by choosing the assortment of products to offer. This decision consists of the number of distinct products to sell n, and the variety and quality attributes of each product j, denoted by (b_j, y_j), respectively. It is assumed that $b_j \in \Re$ and each product is of one of the two quality types, high or low, where

$$y_j = \begin{cases} 1 & \text{if high quality,} \\ 0 & \text{if low quality.} \end{cases}$$

For example, in the case of yogurt, the horizontal attribute b_j could represent the fat content, while the quality attribute y_j could represent regular versus organic options.

Other parameters include the selling price p_j and the purchase cost c_j for product j which depend on the quality level of the product, y_j; we assume p_L and p_H are the prices for the low and high quality products and c_L and c_H are the costs for the low and high quality products. Furthermore, there is a fixed cost K of adding a product to the assortment. All are exogenously determined (with $p_L < p_H$), thus we write p_j and c_j as

$$p_j = (1-y_j)p_L + (y_j)p_H, \quad c_j = (1-y_j)c_L + (y_j)c_H.$$

The distribution function of consumer preferences is denoted as $F(z)$, where $z \in \mathcal{B} \subset \Re$. The utility that a consumer located at z derives from purchasing product j with quality attribute y_j and variety attribute b_j is given by

$$U(z, b_j, y_j) = u(y_j) - p_j - t|b_j - z|, \quad \text{where} \quad u(0) = v \text{ and } u(1) = v + q.$$

Here $u(y_j)$ represents the surplus obtained from purchasing a product of quality level y_j, $v > 0$ is the value of purchasing a product and $q > 0$ represents a quality premium, or additional value obtained from purchasing a high quality product. The last term, $t|b_j - z|$, represents the disutility of a customer located at z when she purchases a product located at b_j, where $t > 0$ can be interpreted as a travel cost for purchasing a product at a non-preferred location.

Coverage distance, l_j: This is the maximum distance that a consumer can be apart from a product j and still obtain a positive utility from the product. The coverage distance is defined as

$$l_j = (1-y_j)l_L + (y_j)l_H; \quad \text{where } l_L = \frac{v - p_L}{t}, \ l_H = \frac{v + q - p_H}{t}.$$

Notice that if $p_L = p_H$ and $q=0$ then the products are identical.

First choice interval, $[b_j^-, b_j^+]$: This interval contains the locations of all consumers who choose product j as a first choice, and is given by

$$b_j^- = \max\left\{ b_j - l_j, \frac{(p_j - u(y_j)) - (p_{j-1} - u(y_{j-1})) + b_j t + b_{j-1} t}{2t} \right\},$$
$$b_j^+ = \min\left\{ b_j + l_j, \frac{(p_{j+1} - u(y_{j+1})) - (p_j - u(y_j)) + b_j t + b_{j+1} t}{2t} \right\}, \quad (1)$$

for $j = 1, \ldots, n$, $b_0 = -\infty$ and $b_{n+1} = +\infty$.

First choice probability, d_j $(\mathbf{b}, \mathbf{y})$: This is the probability that a randomly selected consumer chooses product j from assortment $(\mathbf{b}, \mathbf{y})$ (i.e., a randomly selected consumer belongs to the first-choice interval of product j), and is given by

$$d_j(\mathbf{b}, \mathbf{y}) = \int_{b_j^-}^{b_j^+} f(z)\,dz = F(b_j^+) - F(b_j^-). \quad (2)$$

Assuming that consumer arrivals occur according to a poisson process with rate λ, $D_j(\mathbf{b}, \mathbf{y})$ is a Poisson random variable, denoting the demand for (number of customers who choose) product j as a first choice, with mean $\lambda d_j(\mathbf{b}, \mathbf{y})$. The expected total profit $\Pi(\mathbf{b}, \mathbf{y})$ is the sum of the profits from each product, $\Pi_j(\mathbf{b}, \mathbf{y})$. Thus, the assortment design problem for both the make-to-order and static substitution models is formulated as

$$(\text{Problem } \mathbb{P}) \quad \max_{\{n, (\mathbf{b}, \mathbf{y})\}} \sum_{j=1}^{n} \Pi_j(\mathbf{b}, \mathbf{y}) - n(\mathbf{b}, \mathbf{y})K$$
$$\text{s.t.} \quad d_j(\mathbf{b}, \mathbf{y}), b_j^-(\mathbf{b}, \mathbf{y}), b_j^+(\mathbf{b}, \mathbf{y}) \text{ satisfy (2) and (1).}$$

The exact profit function Π_j differs depending on the problem setting. In the make-to-order case, it is a linear function of the first choice probability, $d_j(\mathbf{b}, \mathbf{y})$, whereas in the make-to-stock with static substitution case, it has been shown to be an increasing convex function of the first choice probability.

2.1. Motivation for further analysis: instances where the optimal solution is unknown

The optimal solution to problem $\mathbb{P}$ has not been found analytically for all possible problem instances. For some cases, the optimal solution is known or has at least been partially characterized [9]. For example, it has been shown that if we know the number of products (n) and their quality levels ($\mathbf{y}$), then the optimal variety attributes of these products ($\mathbf{b}$) can be found by conducting a single variable line-search. Furthermore, conditions exist which further simplify the solution; in particular, for some problem instances it is optimal to only carry products of a single quality level. In this case the optimal assortment is only a function of the location of the first product, b_1. Fig. 1 illustrates a set of problem instances by varying two key parameters, the quality premium (q) and the price premium ($p_H - p_L$) while fixing all other parameters. Observe that some problems, such as (b) lie in the white region; this white region represents a set of problems for which the optimal solution is unknown. On the other hand, point (a) in Fig. 1 is in the gray region; the gray region is a set of problems that satisfy conditions in [9] and are therefore solved analytically. The goal of this research is to close the gap and provide at least an approximate solution to any problem instance in the entire region using a metaheuristic approach.

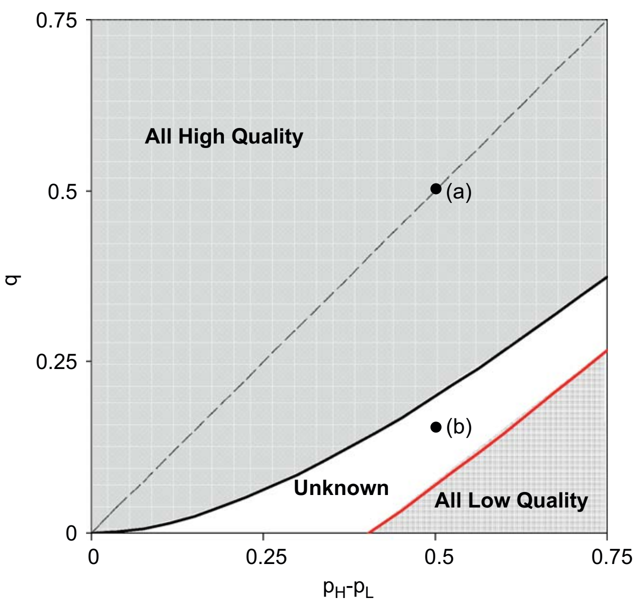

Fig. 1. Illustration of solved and unsolved problems instances for the make-to-order model when K=1, v=2, c_L=c_H=0.5, p_L=1.25, t=4, $\lambda=5$, $F(z)\sim U[0,1]$.

When the distribution of customers is uniform, $F(z)\sim U[0,1]$, the problem can be solved to optimality. Mayorga et al. [10] have shown two ways to find the optimal attributes of mixed quality assortments when products are differentiated both horizontally and vertically. In the make-to-order case, they use a variant of the knapsack, while the static substitution case is solved efficiently using full enumeration. While full enumeration can also be used in the make-to-order case, the variant knapsack formulation returns a solution in a fraction of the time (1 s versus 1 min).

This solution approach, however, cannot be used when the distribution of consumers is unimodal. The inclusion of non-uniform consumer preferences increases the complexity of the problem; the ordering of products matters because the demand for a product does not only depend on its first choice interval, but also on its location. To understand this consider the following example, as illustrated in Fig. 2. In panel (a) on the left, F is a distribution of consumer preferences which is uniform on [0,1]; in panel (b) on the right, $\hat{F}$ is a distribution of consumers which is normal $\sim N(0.5, 0.1)$. Consider two products, one located at b_1=0.3 and the second located at b_2=0.5 with equal coverage distance l=l_1=l_2=0.1. Under F, both products would attain equal first choice probabilities, in particular d_1=d_2=2l=0.2. On the other hand under $\hat{F}$, these two products' first choice probabilities are dependent on b_j, where $\hat{d}_j = \hat{F}(b_j - l) - \hat{F}(b_j + l)$. In this case, $\hat{d}_1 = 0.157$, $\hat{d}_2 = 0.683$. We see that even if only two products should be selected, their optimal locations are not immediately obvious. In the case of n distinct products, we have $n!$ possible orderings. While previous results indicate continuous coverage (which implies that given b_1 we know b_2 through b_n), we are still left with the challenge of locating the first product for each of the $n!$ possible orderings.

3. Results to be used in the design of a metaheuristic approach

This section utilizes previous results from [9] regarding problem $\mathbb{P}$ in order to create bounds on the decision variables which can be used to design more efficient metaheuristic methods. Recall, our goal is to solve problem $\mathbb{P}$, where the exact form of the profit function depends on the substitution behavior. The profit from product j, Π_j is characterized as

$$\Pi_j \in \{\Pi_j^M, \Pi_j^S\},$$

$$\Pi_j^M(d_j(\mathbf{b},\mathbf{y}), y_j) = (p_j - c_j)\lambda d_j(\mathbf{b},\mathbf{y}),$$

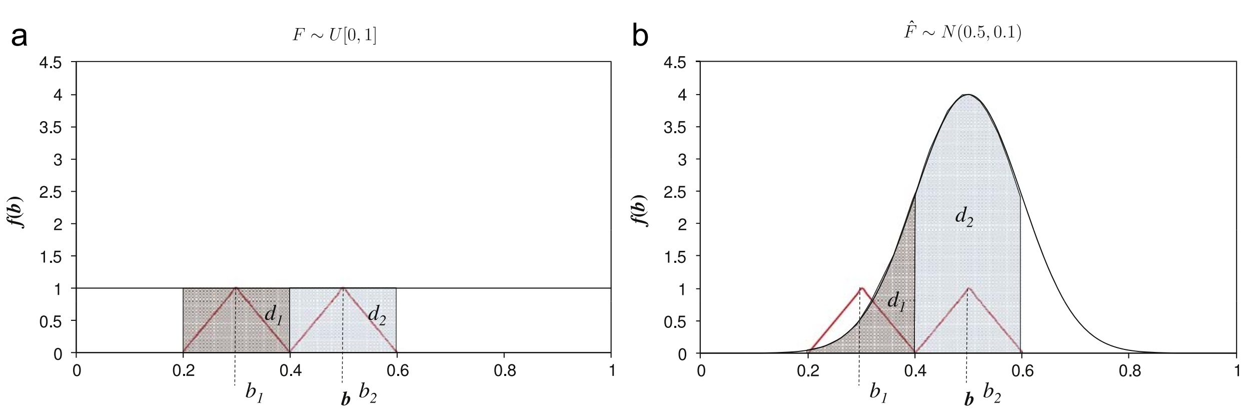

Fig. 2. Illustration of resulting first choice probabilities for two different consumer preference distributions. (a) $\hat{F}\sim U[0,1]$; (b) $F\sim N(0.5, 0.1)$.

$$\Pi_j^S(d_j(\mathbf{b},\mathbf{y}),y_j)=(p_j-c_j)\lambda d_j(\mathbf{b},\mathbf{y})-p_j\phi(z_j)\sqrt{\lambda d_j(\mathbf{b},\mathbf{y})},$$

where $z_j=\Phi^{-1}((p_j-c_j)/p_j)$, ϕ and Φ represent the standard normal probability density function (*PDF*) and cumulative distribution function (*CDF*), respectively. For further detail on how the profit function is derived see [9]. In the make-to-order case (denoted by M), consumers do not substitute while in the case of static substitution (denoted by S), consumers choose not to buy if their most preferred product is not in stock.

In the make-to-order case the profit function is linear while in the make-to-stock case it has been shown to be convex increasing in d_j. However, in either case, the profit for product j depends not only on the location of product j but also on the location of all other products. However, it is known [9] that in optimality the location of products can be restricted in two ways:

(i) $b_j^+-b_j^-=2l_j$, (ii) $b_j^+=b_{j+1}^-$.

These restrictions imply that products do not overlap and there is no gap between them on the variety attribute space. Therefore, we can rewrite $\mathbb{P}$ such that each product's profit is in fact independent of the other:

$$(\text{Problem } \mathbb{P})\quad \max_{\{n,(\mathbf{b},\mathbf{y})\}}\sum_{j=1}^{n}\Pi_j(d_j(b_j,y_j),y_j)-n(\mathbf{b},\mathbf{y})K$$

$$\text{s.t.}\quad d_j(b_j,y_j)=F(b_j+l_j)-F(b_j-l_j)$$
$$b_j=b_{j-1}+l_{j-1}+l_j,\quad j=2,\ldots,n-1.$$

Even with these simplifications, we see that $\mathbb{P}$ remains a non-linear optimization problem with unknown set of variables since n is not known. In particular, to solve the problem we still need to find n, the number of products in the assortment; $\mathbf{y}$, the quality levels of products 1–n; and b_1, the location of the first product. Then b_2–b_n can be found using the second constraint. Next we briefly describe how the solution space can be reduced by finding bounds on b_1, n, and the location of high and low quality products.

When the quality premium exceeds the price premium: In this case $q\geq(p_H-p_L)$, which results in a longer coverage distance from the more profitable product ($l_H\geq l_L$). In such a case the high quality product is said to "cannibalize" the low quality product such that the optimal assortment contains only high quality products [9]. This condition is illustrated in Fig. 1, as the gray area above the diagonal. Recall that in this case the problem can be solved using a single variable line search. While stricter conditions exist (the gray areas below the diagonal in Fig. 1), they may be difficult to find (in the case of unimodal distribution, they can only be found computationally). *Therefore, we restrict our efforts to the set of problems where the high quality products do not cannibalize* ($l_L>l_H$). This means that the optimal assortment may contain multiple quality levels.

Bounds on product location: Products should only be located so that they attain enough profit to cover the fixed cost, that is $b_1\geq\min\{b:\Pi_1(b,y_1)\geq K\}$. This depends on the quality level of product 1, which is an unknown variable. Thus, the bounds on the location of products of type $y\in\{0,1\}$ are given by

$$b_{min}^y=\min\{b:\Pi_j(d_j(b,y),y)\geq K\},$$
$$b_{max}^y=\max\{b:\Pi_j(d_j(b,y),y)\geq K\}.$$

It is useful to develop an absolute minimum location, beyond which no product can be located. We will call this location b_{min} and it represents the first variety attribute (location) where we may place a product that will return non-negative expected profit. Similarly, we can develop an absolute maximum location, b_{max}, representing the last location where we can place a product that will return non-negative expected profit. These bounds are defined as

$$b_{min}=\min\{b_{min}^0,b_{min}^1\},\quad b_{max}=\max\{b_{max}^0,b_{max}^1\}.$$

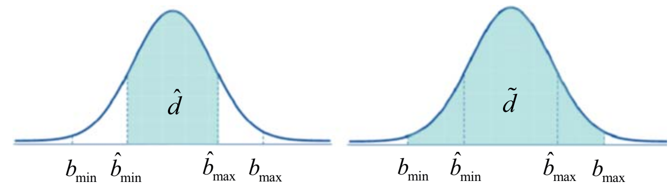

Fig. 3. Product location bounds and profitability crossover points.

Bound on the number of products (N_L, N_H): Using the bounds on product location, we can develop upper bounds on the number of products of each type. Given the coverage distance of products of each quality level, and the minimum and maximum locations, we define N_L and N_H to be the upper bound on the number of low and high quality products where

$$N_L=\left\lfloor\frac{b_{max}-b_{min}}{2l_L}\right\rfloor+1\quad\text{and}\quad N_H=\left\lfloor\frac{b_{max}-b_{min}}{2l_H}\right\rfloor+1.$$

These bounds could be made tighter by letting the b_{max} and b_{min} depend on the quality levels. However, this adds complexity to the problem without gain in efficiency to the metaheuristic methods used.

Low and high quality regions: Recall that when $l_H\geq l_L$, the high quality product cannibalizes, thereby reducing the problem to a single variable line search. As described previously, we only consider case where $l_L>l_H$. In this case the region over which high quality products are profitable is smaller than the region over which low quality products are profitable, since the high quality products have smaller coverage distance and therefore smaller resulting demand. There exist (at least) two locations in each distribution where the more profitable product will switch from low to high (or vice versa). We define these two locations, $\hat{b}_{min}$ and $\hat{b}_{max}$ below

$$\hat{b}_{min}=\min\{b:\Pi_j(d_j(b,0),0)=\Pi_j(d_j(b,1),1)\},$$
$$\hat{b}_{max}=\max\{b:\Pi_j(d_j(b,0),0)=\Pi_j(d_j(b,1),1)\}.$$

It is also of use to define $\tilde{d}$ and $\hat{d}$ as densities given by

$$\tilde{d}=[F(b_{max})-F(b_{min})],\quad \hat{d}=[F(\hat{b}_{max})-F(\hat{b}_{min})].$$

Fig. 3 illustrates an example of these densities and locations.

Regardless of the exact form of the profit function Π_j, problem $\mathbb{P}$ is unique in that it involves several complicating factors: (1) dynamic size, since the number of products to be included is unknown; (2) it is combinatorial problem, in that for a given set of n products we do not know the ordering (which are high quality, which are low quality); (3) continuous variables, in that the exact location of the first product is unknown. Thus, without further assumptions, this problem becomes analytically intractable or too computationally intensive to solve to optimality when F is unimodal. Therefore, we turn to a metaheuristic approach as a way to approximate the optimal solution.

4. Metaheuristics

With the unimodal case presenting a difficult problem to solve optimally, we employ a series of metaheuristic techniques to provide high quality solutions in a relatively small amount of time. Three methods are investigated: a genetic algorithm (GA), simulated annealing (SA), and a tabu search (TS). The goal of this exercise is not only to provide an efficient method of generating solutions to this class of problems, but also to study the effectiveness of each of these techniques for solving these types of problems. We begin by describing the implementation and design decisions for the study and for each of the metaheuristics.

4.1. Common elements

The metaheuristic approaches share several common elements. These include the solution representation, the fitness function, and the upper bound on profit.

Solution representation: The solution representation is illustrated in Fig. 4. Each representation consists of N_H+N_L elements and a real number. The real number on [0,1], known as the *offset*, is used to compute $b_1=b_{min}+offset$.

The remaining N_H+N_L elements each represent a potential product in the solution, with the total number of elements equal to the sum of the upper bound on low quality products in the assortment, N_L, and the upper bound on high quality products in the assortment, N_H. Each element designates a product as high or low quality and contains a random number on the interval (0,1) which serves as a random key for sorting purposes. Random keys have been used frequently and were first proposed by Bean in 1994 [1]. We discuss the decoding of the representation in the following, as it is inherently related to computing the objective function.

Objective value: In this research, our goal to maximize expected profit. Each solution is decoded by a function which calculates the expected profit for the given solution, as follows:

1. The N_H+N_L elements are sorted by the random key field (from smallest to largest).
2. *Current Location*=(b_{min}+*offset*-$l_{[1]}$) and *Count*=0.
3. For i=1 to N_H+N_L:
   (a) Consider the [i]th sorted element.
   (b) Determine the quality level of the product and calculate $\Pi_{[i]}$, the expected profit for the product at *Current Location*+$l_{[i]}$.
   (c) If $\Pi_{[i]} \geq K$ the product is added to the assortment and the total expected profit is updated. b_{Count}=*Current Location*+$l_{[i]}$, *Current Location*=*Current Location*+2$l_{[i]}$, and *Count*=*Count*+1
   (d) Else, If *Count*=0, then: *Current Location*=*Current Location*+2$l_{[i]}$.

Upper bound: As an optimal solution is not available in the general unimodal case, we employ the use of an upper bound on expected profit for solution quality evaluation. To compute the upper bound, we divide the customer preference region into up to two areas: an area in which high quality products will be more profitable, and an area in which only low quality products will be profitable. We can then derive an upper bound on expected profit for each region and the sum will provide the upper bound for our problem. For each region, we assume that the entire density can be captured by placing products, so the gross revenue is equal to the density multiplied by the profit margin for the respective item. We pay the fixed cost for each product that may be placed in that region. Because this method allows for partial products that will only require a fraction of the fixed cost, the bound will not be tight to the optimal solution. The bounds will depend on the problem instance. Consider the following three cases:

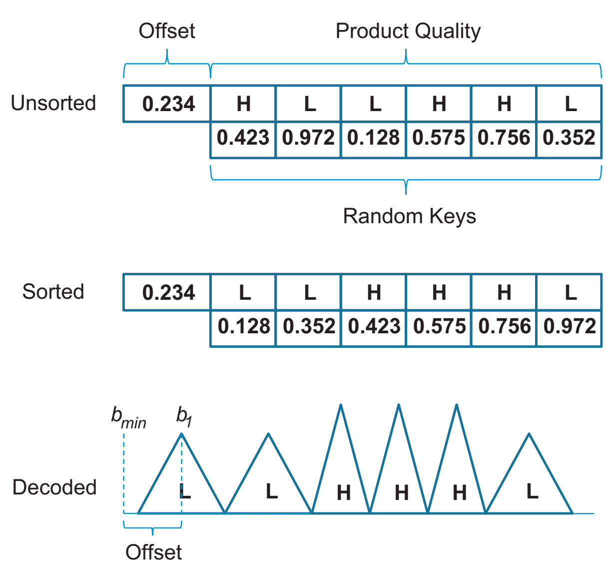

Fig. 4. Sample solution representation.

Case unmixed 1: $b_{min}=b^1_{min}$. This implies that a high quality product is more profitable than a low quality product at the same location, anywhere in the feasible location region. Thus the optimal assortment will consist of only high quality products. An upper bound on profit is given as follows:

$$UB_1 = \Pi_j(\tilde{d},1) - \left\lfloor \frac{b_{max}-b_{min}}{2l_H} \right\rfloor K.$$

Case unmixed 2: $b_{min}=b^0_{min}$ and $\hat{b}_{min} < b_{min}$. In this case the optimal assortment will consist of only low quality products. An upper bound is given by

$$UB_2 = \Pi_j(\tilde{d},0) - \left\lfloor \frac{b_{max}-b_{min}}{2l_L} \right\rfloor K.$$

Case mixed: otherwise. In this case the optimal assortment may contain both high and low quality products, but high quality products are only profitable in the region bounded by $(\hat{b}_{min}, \hat{b}_{max})$. Thus the bound is given by

$$UB_3 = \Pi_j(\hat{d},1) - \left\lfloor \frac{\hat{b}_{max}-\hat{b}_{min}}{2l_H} \right\rfloor K + \Pi_j((\tilde{d}-\hat{d}),0) - \left(\left\lfloor \frac{\hat{b}_{min}-\hat{b}_{min}}{2l_L} \right\rfloor + \left\lfloor \frac{b_{max}-\hat{b}_{max}}{2l_L} \right\rfloor \right) K.$$

4.2. Genetic algorithms

Genetic algorithms (GA) are population based, evolutionary metaheuristics, whose use for combinatorial problems is described in Goldberg [4]. A population consists of chromosomes, each of which maps to a point in the solution space. We use the solution representation discussed in Section 4.1 as our chromosome. Once an initial population is established, future generations are produced through the use of these operators:

- *Elite reproduction*: Chromosomes may be passed untouched to the next generation through the use of elite reproduction. Typically, the "most fit" chromosomes (as judged by a fitness function) will be passed on, with the number determined by the implementation. In elite reproduction, the best X% of the current generation is copied, unaltered, to the next generation. The fixed percentage X is often set at 20%, as in Bean [1].
- *Descendants*: Descendant chromosomes are produced through the "mating" of two parent chromosomes. Our implementation uses a single point crossover operator to produce descendants. Single point crossover produces two offspring from two parents. One descendant will contain the genetic information from one parent up to a crossover point and the genetic information from the other parent after the crossover point. The second descendant will contain the complementary genetic information from each of the parents. Both offspring are retained for the future generation. The crossover operation is illustrated in Fig. 5.

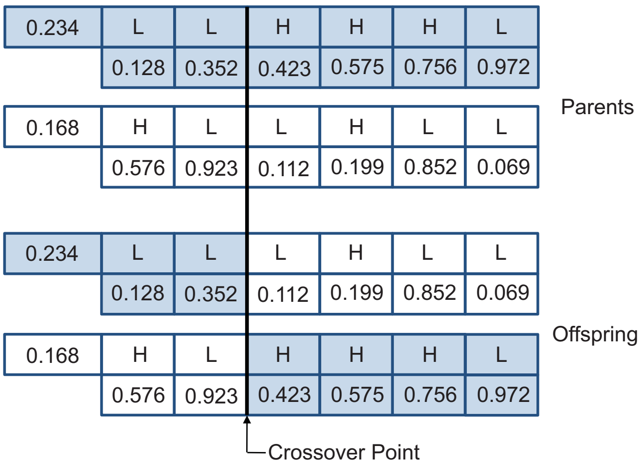

Fig. 5. Illustration of the crossover operator.

- *Immigration*: Immigration is the process of introducing entirely new chromosomes to the population. In our case, these chromosomes are generated randomly.

The genetic algorithm contains a population of 100, which is initialized with a random starting population. Following populations are built using 20% elite reproduction, 70% descendants, and 10% immigration. Parents are chosen randomly from the entire previous population and reproduction is performed with a single point crossover with a random crossover point. One thousand generations are evaluated. All parameters were chosen empirically based on a subset of initial test cases. For example, to enable a *fair* comparison between methods, we needed to establish a fixed number of solution evaluations over all three methods. To establish this limit we found the number of solutions needed to converge over a subset of test cases for all three methods and increased this number by some factor to err on the side of caution.

4.3. *Simulated annealing*

Simulated annealing (SA) is a path based metaheuristic first applied to optimization by Kirkpatrick in 1983 [5]. The method is intended to approximate the natural process in which metal is annealed, and employs the Metropolis algorithm. The metaheuristic begins with an initial solution and a "temperature" value. New solutions are generated by performing a small modification to the previous solution. New solutions that show an improvement in objective value are always accepted, and solutions which produce a inferior objective value are accepted with probability $e^{\Delta\varepsilon/T}$, where $\Delta\varepsilon$ is the change in objective value and T is the current temperature of the system. The temperature is lowered once "stability" is reached at a temperature level (determined empirically) and the algorithm is finished after the solution becomes stable over multiple temperature levels. The cooling scheme used to lower the temperature is geometric, with the updated temperature set to 90% of the current temperature.

The simulated annealing process begins with a random starting solution. The initial temperature is set according to the characteristics of the individual problem instance. New solutions are generated by either randomly modifying the offset (10% probability) or changing the quality level of a randomly selected product (90% probability). The offset modification is performed by generating a $(-1,1)$ random number which is multiplied by σ (the standard deviation of customer preference) and added to the previous offset. For comparison purposes to the other methods, a limit of 100,000 solutions was enforced.

4.4. *Tabu search*

Tabu search (TS) was proposed and applied to optimization problems by Glover in 1986 [3]. Like simulated annealing, tabu search is a path based metaheuristic. An initial solution is generated, as well as a list of "neighbor" solutions. Neighbor solutions are defined as a group of solutions that border the original solution and usually involve a single perturbation of the original solution. A tabu list is established, which is a listing of recently visited solutions (the length of the list is determined by the implementer). For each "move" in the algorithm, the tabu search generates the list of neighbor solutions and moves to the neighbor with the most desirable objective value that is not present on the tabu list.

The tabu search begins with a random starting solution. The tabu list consists of 10 elements, and any solution with the same objective function is considered identical. In other words, our tabu list consists of a set of objective values rather than a set of solutions. We do this for two reasons. First, because of the representation many solutions could yield the same objective function; second, our solution representation contains a continuous variable so that it is unlikely that the exact same solution be visited again; thus, a tabu list consisting of full solution representations would be very inefficient. The search consists of 10,000 updates, with N_H+N_L+1 neighbors generated by randomly modifying the offset and changing the quality level of each product individually. The offset modification is performed by generating a $(-1,1)$ random number which is multiplied by σ and added to the previous offset. Both tabu search and simulated annealing use the same perturbations. Simulated annealing allows only one solution to be considered per move, and tabu search considers the entire neighborhood of solutions and selects the best non-tabu move.

5. Computational experiments

The three methods were each evaluated computationally over a series of 656 test cases. The cases are divided into three p_L values, which each contain two K values, and then a range of p_H and q values are presented which result in l_H values of interest (recall we are interested in cases such that $l_H < l_L$), all other parameters are fixed. For example, for p_L=1.5, parameters are such that l_L=0.25; thus, we vary p_H and q such that $p_H - p_L < q$, resulting in $l_H < 0.25$. A detailed list of test cases is available at http://people.clemson.edu/~mayorga/papers/CORdata.pdf. An upper bound for each case was computed off-line. Each method was implemented in C++ and compiled with *Microsoft Visual Studio 2008*. Fifty replications of each test case were evaluated using *Condor*, a high throughput grid computing solution. The Mersenne Twister pseudorandom number generator was used and each replication of each run used a pre-generated seed that ensured no overlapping of random number streams. Results were computed by measuring deviation from the bound for each replication, and averaging the results for each case over the 50 replications. Computational time for a single case on a P4 3.20 GHz PC with 2 GB of RAM was < 1 s.

6. Results

The aggregated results are presented in Table 1. *Bold* entries show the best value in each row. Each row represents a single test case or an aggregation of a class of test cases. These results show similar performance by the GA and TS, and both are well ahead of SA. Each method shows low variance (in the order of 10^{-6}) and as a result we are satisfied with the convergence of each method.

Table 1
Aggregated results.

	% Deviation from bound [Avg (Min, Max)]		
	GA	TS	SA
All	**2.590% (0.018%, 26.070%)**	2.629% (0.022%, 26.616%)	3.487% (0.135%, 26.160%)
Mixed	**3.471% (0.018%, 26.070%)**	3.510% (0.022%, 26.616%)	4.667% (0.197%, 26.160%)
Unmixed	1.099% (0.059%, 4.456%)	**1.040% (0.043%, 4.280%)**	1.357% (0.135%, 4.694%)

Table 2
Case results.

p_L	K	Bound	% Deviation from bound [Avg (Min, Max)]		
			GA	TS	SA
1.5	0.2	All	**1.846% (0.364%, 14.829%)**	1.962% (0.367%, 15.825%)	2.774% (0.605%, 15.111%)
		Mixed	**2.321% (0.522%, 14.829%)**	2.485% (0.758%, 15.825%)	3.337% (1.017%, 15.111%)
		Unmixed	0.535% (0.364%, 0.702%)	**0.519% (0.367%, 0.716%)**	1.221% (0.605%, 2.224%)
	1.0	All	**9.237% (2.498%, 26.070%)**	9.426% (2.214%, 26.615%)	9.506% (2.503%, 26.160%)
		Mixed	**13.074% (5.074%, 26.070%)**	13.417% (5.433%, 26.615%)	13.380% (5.765%, 26.160%)
		Unmixed	3.570% (2.498%, 4.456%)	**3.529% (2.214%, 4.280%)**	3.782% (2.503%, 4.694%)
50	10	All	0.577% (0.170%, 7.622%)	**0.483% (0.088%, 2.514%)**	2.451% (0.153%, 12.614%)
		Mixed	0.803% (0.295%, 7.622%)	**0.691% (0.337%, 2.514%)**	3.843% (0.694%, 12.614%)
		Unmixed	0.255% (0.170%, 0.345%)	**0.187% (0.088%, 0.321%)**	0.472% (0.153%, 1.071%)
	50	All	**2.803% (0.599%, 25.061%)**	2.843% (0.504%, 25.820%)	3.937% (0.619%, 25.189%)
		Mixed	**4.177% (1.714%, 25.061%)**	4.350% (1.552%, 25.820%)	6.122% (2.256%, 25.189%)
		Unmixed	1.121% (0.599%, 1.750%)	**0.997% (0.504%, 1.736%)**	1.262% (0.619%, 1.877%)
60	1	All	0.171% (0.018%, 0.327%)	**0.168% (0.021%, 0.327%)**	0.885% (0.134%, 3.301%)
		Mixed	**0.207% (0.018%, 0.327%)**	0.208% (0.021%, 0.327%)	1.103% (0.196%, 3.301%)
		Unmixed	0.084% (0.058%, 0.110%)	**0.071% (0.042%, 0.099%)**	0.351% (0.134%, 0.990%)
	5	All	0.942% (0.298%, 2.020%)	**0.929% (0.251%, 2.072%)**	1.407%(0.376%, 3.780%)
		Mixed	**1.176% (0.554%, 2.020%)**	1.184% (0.588%, 2.072%)	1.779% (0.763%, 3.780%)
		Unmixed	0.441% (0.298%, 0.600%)	**0.383% (0.251%, 0.569%)**	0.611% (0.376%, 1.002%)

Subdividing the tests cases gives the results as presented in Table 2. Here we can observe the relative strengths of both the GA and the TS. The GA offers increased performance in mixed cases (Cases 1 and 2) and generally in the class of problems that we consider to be "hard" (the mixed cases are a member of this class). Harder problems tend to occur in the situations where there is a small price premium or quality premium between products. Hard problems are also those in which the K value is relatively high (problems in which it is more difficult to obtain positive profit from each product). The TS method, by contrast, excels at problems in which the resulting assortment is unmixed. These are problems in which the assortment is relatively easy to find and the difficulty is in finding the correct offset.

Box plots of the results are shown in Fig. 6. Inspection reveals that these results are not normally distributed. Non-parametric statistical testing confirms the observational results above. A statistical ranking of the three methods is presented in Table 3. Non-parametric techniques are used, as the results are skewed towards the upper bound, thereby eliminating the normality assumption. Friedman's test revealed significant statistical differences between the methods, and then a multiple comparison test was used to rank the methods for each case. We see that the results in Tables 2 and 3 are consistent.

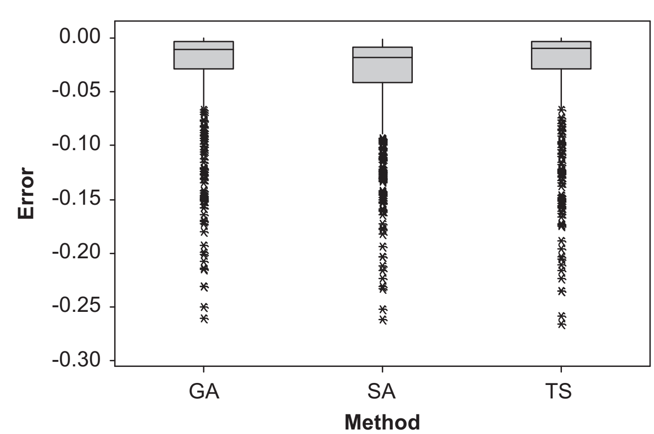

Fig. 6. Boxplot of percent deviation from bound.

7. Conclusions

Simulated annealing can be eliminated as an effective tool for solving our problem cases. SA performed at an inferior level in each of the cases tested, suggesting a poor fit for these problem cases and for the multiple quality assortment problem in general. We speculate that the performance of the SA suffers from only considering a single solution at each iteration, but we have not tested this specific assertion.

The genetic algorithm and tabu search perform better than simulated annealing in general in these problems, as demonstrated by our statistical analysis. Relative to the upper bound, GA and TS can perform within 0.018% and 0.021%, respectively, on some problems. While the GA performs slightly better, averaging over all cases, statistical ranking shows that the difference is not significant. On the other hand, the TS does out-perform the GA in

Table 3
Statistical ranking.

p_L	K	Bound	GA	TS	SA
1.5	0.2	All	**1**	2	3
		Mixed	**1**	2	3
		Unmixed	**1**	**1**	3
	1.0	All	**1**	2	2
		Mixed	**1**	2	2
		Unmixed	**1**	**1**	3
50	10	All	2	**1**	3
		Mixed	**1**	**1**	3
		Unmixed	2	**1**	3
	50	All	2	**1**	3
		Mixed	**1**	**1**	3
		Unmixed	2	**1**	3
60	1	All	**1**	**1**	3
		Mixed	**1**	**1**	3
		Unmixed	2	**1**	3
	5	All	**1**	**1**	3
		Mixed	**1**	**1**	3
		Unmixed	2	**1**	3
		Case			
		All	**1**	**1**	3
		Mixed	**1**	2	3
		Unmixed	2	**1**	3

more case-by-case comparisons. Therefore, without pre-classifying the problem (to know if the assortment will be mixed or unmixed), both GA and TS methods are recommended for solving the assortment planning problem.

Previous work with these problems [9] has suggested that they may be pre-classified by instance type and then solved. For example, as shown in Fig. 1, problems in the gray region will contain a single quality type (unmixed case), while problems in the white region may contain multiple quality types (mixed case). The unmixed case can be solved using several single variable line searches.

For mixed problem instances, the genetic algorithm is the preferred method for solving assortment problems. The GA offers increased performance for these ''hard'' problems. On the other hand, for the unmixed cases, the TS method was the dominant method. This advantage seems to stem from the TS method's ability to modulate an existing offset, as opposed to generating entirely new random offsets, as the GA does. Also, TS considers changing the quality level of any product and only changes the offset when it will provide the largest gain in the objective value. It seems reasonable to suggest that adding an offset mutation to the GA method would allow for better performance against the TS in these cases.

The computational work necessary to pre-classify problems (as mixed or unmixed) is non-trivial. Fortunately, the use of these metaheuristic methods to solve assortment problems eliminates the need to pre-classify the problems. Thus, an interesting side effect of our work shows that since these methods work so quickly over all problem types, there is no need to pre-classify problems when the only end goal is to find a solution to the problem. In this case both GA and TS methods are highly competitive.

Additionally, it is worth making a note on the quality of the bounds used in this evaluation. The bounds, by nature, will underpay the fixed cost of the generated assortment. As such, the gap between the optimal solution and the bound on profit grows greatly with K, and becomes even larger as the marginal-profit-to-fixed-cost-ratio grows. This relationship is illustrated in our results and should be considered more of a statement on the quality of the bound than a statement on the quality of the results. However, no tighter bound is known at this time for these cases.

We have shown that metaheuristic techniques can be efficiently and effectively used to approximate solutions to difficult to solve assortment planning problems. Thus researchers may look to such methods as more pragmatic retail models are developed. One possible extension is generalizing the assumption of consumer preferences beyond the unimodal distribution. For example, if consumer preferences are bi-modal the optimal assortment may contain overlapping products, thus the location of each product needs to be determined and the solution representation would have to be altered accordingly. Another possible extension is to consider the dynamic substitution environment, where consumers will attempt to substitute for their most preferred product if it is out of stock. In this case the assortment and inventory problem do not decouple, making the problem analytically intractable. Not only is a new solution representation necessary, but also a more complex objective function.

Having established the viability of metaheuristics as a solution approach, we are currently working on developing a special purpose genetic algorithm to solve assortment problems in which the products may have overlapping coverage. This occurs when consumer preference for quality is non-homogeneous. In this case, since properties of the optimal assortment are difficult to characterize, our goal is to use the GA to generate solutions which will give us insights as to the structure of the optimal policy.

References

[1] Bean JC. Genetic algorithms and random keys for sequencing and optimization. ORSA Journal on Computing 1994;6(2):154–60.
[2] Gaur V, Honhon D. Assortment planning and inventory decisions under a locational choice model. Management Science 2006;52(10):1528–43.
[3] Glover F. Future paths for integer programming and links to artificial intelligence. Computers and Operations Research 1986;13:533–49.
[4] Goldberg DE. Genetic algorithms in search, optimization and machine learning. Reading, MA: Addison-Wesley; 1989.
[5] Kirkpatrick S, Gelatt C, Vecci M. Optimization by simulated annealing. Science 1986;220(4598):671–80.
[6] Lancaster K. A new approach to consumer theory. Journal of Political Economy 1966;74:132–57.
[7] Mayorga ME. Essays on consumer heterogeneity and optimal stochastic resource allocation. PhD dissertation, University of California, Berkeley; 2006.
[8] Mayorga ME. Optimal retailer assortment and stocking decisions. In: Proceedings of the 2007 industrial engineering research conference, Nashville, TN; 2007.
[9] Mayorga ME, Ahn HS, Aydin G. Optimal assortment and stocking decisions with multiple quality levels. Working Paper, February 2008. Available online at 〈http://people.clemson.edu/~mayorga/papers/NRL.pdf〉.
[10] Mayorga ME, Kurz ME, McElreath MH. Optimal attributes of mixed quality assortments. In: Proceedings of the 2008 industrial engineering research conference, Vancouver, Canada, BC; 2008.