Available online at www.sciencedirect.com
ScienceDirect
European Journal of Operational Research 176 (2007) 498–520
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/ejor

O.R. Applications

Optimal product design using a colony of virtual ants

M. David Albritton [a,*], Patrick R. McMullen [b,1]

[a] Northern Arizona University, College of Business Administration, P.O. Box 15066, Flagstaff, AZ 86011, USA
[b] Wake Forest University, Babcock Graduate School of Management, Room 2109, Winston-Salem, NC 27109, USA

Received 27 January 2005; accepted 17 June 2005
Available online 16 September 2005

Abstract

The optimal product design problem, where the "best" mix of product features are formulated into an ideal offering, is formulated using ant colony optimization (ACO). Here, algorithms based on the behavior of social insects are applied to a consumer decision model designed to guide new product decisions and to allow planning and evaluation of product offering scenarios.

ACO heuristics are efficient at searching through a vast decision space and are extremely flexible when model inputs continuously change. When compared to complete enumeration of all possible solutions, ACO is found to generate near-optimal results for this problem.

Prior research has focused primarily on optimal product planning using consumer preference data from a single point in time. Extant literature suggests these formulations are overly simplistic, as a consumer's level of preference for a product is affected by past experience and prior choices. This application models consumer preferences as evolutionary, shifting over time.

Keywords: Combinatorial optimization; Heuristics; Product design/planning; Ant colony optimization (ACO); Swarm intelligence (SI)

1. Introduction

The design of a firm's products, as well as its product positioning and product mix selection, is critical from a strategic perspective because of its considerable impact on long-term firm performance. The product design (PD) problem involves determining the best (and preferably, optimal) mix of individual characteristics for product or service offerings. These offerings are optimized to include those characteristics that are best suited

* Corresponding author. Tel.: +1 928 523 7390; fax: +1 928 523 7331.
E-mail addresses: david.albritton@nau.edu (M.D. Albritton), patrick.mcmullen@mba.wfu.edu (P.R. McMullen).
[1] Tel.: +1 336 758 4244.

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2005.06.042

to consumer preferences. Because of a direct link to consumer demand, these ''optimal'' offerings are assumed best able to perform effectively in a competitive environment and are in the best position to generate profitability for the firm. The PD problem is inherently complex because many different product combinations must be evaluated and considered in the context of the firm's bigger picture. This problem requires the firm to fulfill customer expectation and need for variety while not endangering financial performance by overextending the firm with too many products or compromising the manufacturing efficiency of the enterprise.

New product introductions are typically very expensive to design and market (Urban and Hauser, 1993) and historically have an extremely high rate of failure (Business Week, 1993). Quelch and Kenny (1994) warn that inappropriate expansion of a product line can deteriorate a brand's image and negatively affect partner relationships. Furthermore, product extensions can also result in the cannibalization of existing products (Green and Krieger, 1987). To help minimize these risks, market researchers have developed sophisticated methods to identify and evaluate new product concepts prior to their introduction.

Since its first application in the early 1970s, problems dealing with the optimization of design features have been popular in the marketing and management literature (Shocker and Srinivasan, 1974, 1979; Zufryden, 1977; Green and Krieger, 1985, 1989; Kaul and Rao, 1995). The PD problem is typically formulated to follow the marketing concept, as consumer perceptions about product offerings constitute the critical inputs for design plans. Griffin and Hauser (1993) describe the ''voice of the consumer'', where customer needs are linked to product design attributes that facilitate engineering, manufacturing and R&D decision-making. Here, new product identification and evaluation is conducted with a sharp focus on consumer expectation using methods such as conjoint analysis or multi-dimensional scaling (Green and Srinivasan, 1978, 1990).

The linkages between product characteristics and attributes, customer preferences, and product selection have been conceptualized into a framework where design decisions can be made (Kaul and Rao, 1995; among others). Finding the optimal mix of product features or characteristics places the firm in a position to maximize customer choice (Green et al., 1981), usually in the form of increased market share (Kohli and Sukumar, 1990), maximized profits (Hauser and Simmie, 1981), minimized product line cannibalization (Dobson and Kalish, 1993), maximum buyer's welfare (McBride and Zufryden, 1988), or maximized seller's return (Dobson and Kalish, 1988).

Unfortunately, product optimizations are problematic. As the number of product features under consideration grows, the problem is increasingly difficult to manage. For example, an overly simplistic product with only seven features and three potential choices for each feature equates to 3^7 or 2187 potential product offerings; one additional feature to the mix, increases the number of unique offerings to 3^8 or 6561. Here, problem complexity increases at an exponential rate. The product design problem is NP-hard, meaning ''the time needed to solve the problem increases at a much higher rate than the increase in the size of the problem'' (Kaul and Rao, 1995, p. 315). Given the computational difficulty of large-scale problems like PD, analytical solutions are often impractical to ascertain. With problems of a particularly large magnitude, analytical solutions can be virtually impossible to ascertain (Kohli and Krishnamurti, 1989).

Over time, two distinct schools of thought regarding product optimizations have evolved (Dobson and Kalish, 1993). The first area includes research endeavors using formulations based upon pure optimization where every possible permutation is considered; the second area includes research where heuristic procedures are used as a tool to narrow possible combinations. Heuristic procedures are suggested as a means to manage the decision space of problems that are computationally NP-hard (Green and Krieger, 1985). Heuristics such as greedy interchange (Dobson and Kalish, 1993), beam search (Nair et al., 1995), dynamic-programming (Kohli and Krishnamurti, 1987), and genetic algorithms have been applied to the PD (Balakrishnan and Jacob, 1996) and PLD (Alexouda and Paparrizos, 2001) problems.

The use of heuristic procedures allows the product design problem to maintain richness of information, without having to overly constrain the problem by limiting complexity. The extant literature includes several instances where the use of heuristics is attacked (Zufryden, 1977; McBride and Zufryden, 1988) suggesting, for example, that modern computing can handle any problem of any size and that the extra time needed to find the true optimum is warranted. Unfortunately, these pure optimization formulations are often unrealistic and must sacrifice problem complexity in order to converge upon a true optimal solution.

The perspective of the authors mirror Green and Krieger (1989) and Kohli and Krishnamurti (1989) in that pure optimization using complete enumeration is the preferable choice when the problem is sufficiently small, with a manageable degree of magnitude. For large-scale problems, heuristics offer an alternative means to reach near-optimal solutions, without the need to overly constrain problem formulations. As computing environments become more sophisticated and problem size becomes increasingly more manageable, this enhanced level of sophistication can be used logically to make problem formulations more realistic and complex. Rather than maintaining overly simplistic formulations and assumptions for the sake of mathematical completeness, well-developed heuristics allow modelers to include new variables in formulations and still maintain computational efficiency.

Because the product design problem is NP-hard, a novel heuristic procedure, called ant colony optimization (ACO) is employed to manage the vast decision space. To date, ACO has not been applied to the PD problem. Here, the PD problem is operationalized using conjoint analysis techniques (Green and Srinivasan, 1978, 1990; Green and Krieger, 1989) with simulated consumer preference data generated and modeled via Monte Carlo simulation.

The purpose of this study are threefold: (1) to address product design in the context of a consumer decision model that can link product characteristics and costs, as well as form a connection between product choice and firm revenue, (2) to describe the ACO heuristic method and to introduce this methodology to PD problems of increased complexity, and (3) to expand upon the extant ACO literature by testing certain core methodological assumptions.

This study introduces a dynamic approach to the PD problem by including a temporal component that models changing consumer perceptions and reveals the consequences of this impact on near optimal product designs. Previous work has considered the PD problem in the context of a static environment, where consumer perceptions and attributes are treated as fixed and do not change.

This particular problem formulation also presents a novel objective function, or purpose for PD search, to maximize the net present value (NPV) of market share over a given span of time. Kaul and Rao (1995, p. 317)) support this objective function, stating that there exists a "clear opportunity to develop dynamic models of product positioning and design that take state dependence of choice into consideration", where consumption of a product in the past may affect future attribute perceptions and where prior choice may affect future preference for a given product. This incremental step forward contributes to the sophistication of PD problem modeling, as product consumption affects perception, which then affects future choice decisions.

2. Product design problem and optimization

2.1. Approach to the problem and description of data

In framing the PD problem, perceptual attributes are abstracted from consumers on product characteristics through conjoint analysis, as well as via marketing mix components that may directly or indirectly affect consumer perception. Fig. 1 presents a model of consumer decision-making. It is included here because PD can focus upon a wide variety of optimization areas. For example, PD decisions can focus upon maximizing revenue generation, can focus upon minimizing product costs, or can focus upon some combination of both. The extant marketing and psychology theory suggests that consumers create preferences based on their perceptions and will inevitably make purchase decisions based on their preference for a given product or service (see Kaul and Rao, 1995 for a review of the literature). These choices directly affect revenue and drive firm profit. Conversely, PD can be based upon a manufacturing perspective, looking

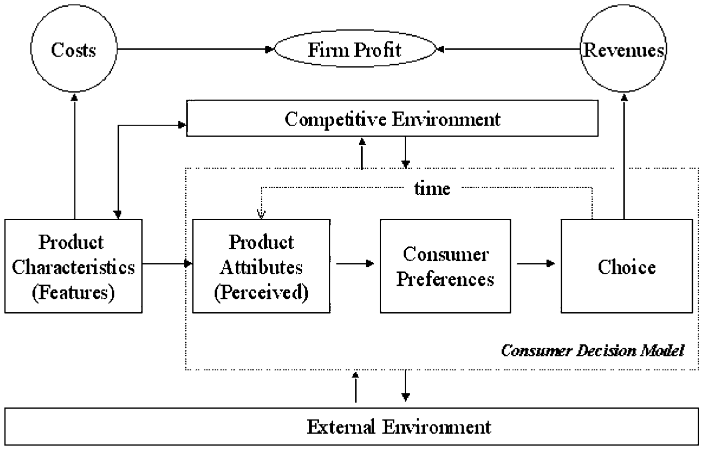

Fig. 1. Model of consumer decision making. *Source*: Brunswik (1955), Hauser and Simmie (1981), Moore and Winer (1987) and Kaul and Rao (1995).

to reveal the optimal mix of product characteristics to lessen the cost of production. For this particular endeavor, we focus primarily upon the marketing aspect of PD, modeling consumer choice with the objective to maximize market share.

The causal relationships between product characteristics, perceptions and preferences have been empirically tested and supported in the literature (Hauser and Simmie, 1981; Kaul and Rao, 1995), with the relationships between each level of abstraction from physical to perceptual characteristics examined. Consequently, this study treats these relationships as given. We, instead, focus upon modeling the process in the hopes of finding the optimal (or near-optimal) mix of attributes for a product offering given changing consumer perceptions and, accordingly, changes in choice over time.

2.2. Consumer perception

According to Moore and Winer (1987), a customer's selection of a product is largely based on his or her perception of the product attributes, along with accompanying preferences for attributes, choice mechanisms, and product price. Additionally, Green and Krieger (1989, p. 128) state, "the conceptual basis of optimal product design rests on the assumption that choice for a product can be related to the buyer's perception and preferences for the product's underlying attribute levels, relative to competing products".

Consumer perception, or preference, in the form of multivariate product information, is the critical input variable for this analysis. Consumer preference data is traditionally gathered via conjoint analysis, multidimensional scaling or componential segmentation techniques (Green and Wind, 1975; Green and Srinivasan, 1990). Once gathered, consumer preferences are organized into consumer utility matrices (part-worths matrices with conjoint data), exhibiting each consumer's individual preference for each level of a product attribute.

For example, Fishken (1988) presents a multivariate analysis of pizza toppings that was conducted for a premium frozen pizza manufacturer. Products were evaluated on twenty attributes (including the amount of cheese, the amount of meat, variations in sauce, and the inclusion of spices) with "overall liking" or preference noted. In this analysis, the researchers sought to identify optimal levels of pizza toppings to satisfy consumer preference. Here, the consumer utility matrix would show individual respondent preference for every level of each attribute (e.g., preferences for 4 oz., 6 oz., or 8 oz. of meat topping).

Unlike Fishken's approach, this effort does not use data collected from consumers on preference levels. Instead, a conjoint analysis utility matrix with consumer preference data is replicated using an algorithmic approach where preference levels are randomly generated within a pre-specified upper and lower bound limit. Using simulated input data is a controversial practice as many researchers question the external validity of such findings. While we appreciate these concerns, simulated consumer data offers benefits over real-world data including enhanced flexibility and the statistical control needed to test our hypotheses. This research endeavor follows the direction of Hooker (1995), where replicated input data is considered necessary and useful for the purposes of testing a combinatorial optimization heuristic.

2.3. Consumer choice

Choice is an output variable in the consumer decision model, directly affecting revenue streams, and indirectly firm profit. Choice can be modeled as deterministic; the assumption being that each consumer has a utility function and will always choose the product with the highest overall utility. Consumer choice can also be modeled probabilistically combining deterministic components with random components. Decision-making is subject to situational constraints that are beyond the marketer's control such as time or money, consequently indiscriminate or uncontrollable components should be included in PD models. Kaul and Rao (1995, p. 307) state, "probabilistic models have increased in popularity" as "they better describe actual consumer behavior". Lehmann (1971) argues that consumer choice, and resultant market share, should be modeled probabilistically as the relative proximity of a product to a customer's ideal preference point.

We follow the work of Lehmann (1971) and Kaul and Rao (1995) using the typical probabilistic model approach. Here, the consumer choice process is modeled as an additive function of individual product attributes with a component of randomness included. After a consumer's preferences for specific individual product attributes are satisfied, the individual attribute satisfaction levels are combined into an overall measure. A consumer will "choose" the product based on the perceived additive part-worths of each attribute, or stated differently, the consumer chooses a product with the highest utility.

This problem is formulated to find that mix of product attributes that generates the maximum possible market share, or attracts the largest number of individual customers in terms of highest utility of satisfaction. Using the aforementioned pizza example, consumers evaluate each individual part of their choice to determine if the pizza has the right levels of cheese, meat, and spices. This utility data is then used to model choice, with a degree of environmental randomness included in the model.

The consumer decision process relates prior choice to future preference. Here, consumer preferences evolve over time, with future preference treated as a function of past preference, with a probabilistic element to include randomness (this process is detailed in Section 3). Consequently, consumer choice evolves as preferences change. Those products that are the most "robust" in terms of longevity, meaning products that remain closest in perceptual space to the largest number of consumer "ideal points" over several points in time, will be given preferential advantage toward the model's objective function, defined as the NPV of a share-of-choices.

Perceptual mapping is a useful tool to depict product positioning, and to demonstrate how consumers distinguish between products on similar dimensions for choice. Conjoint techniques will typically not incorporate perceptual mapping due to the large number of product attributes to be mapped and as each attribute does not always have levels which can be easily defined on a continuous distribution. Nonetheless, a perceptual map is useful for illustrative purposes. Fig. 2 exhibits a perceptual map with four customer inputs and two key product dimensions. With this simplified perspective, consumers describe the product as "ideal" on two dimensions, price and performance. Product offerings can be designed to fall along these dimensions within the area of maximum consumers' preference. The arrows in this diagram show a drift of consumer perception over time, as consumer preferences will inevitably change (here, dashed circles shift to become solid circles over time). For example, a consumer increasingly demands a higher performance computer at a lower price. Here, the map graphs price and performance as a function of shifting consumer preference. Referring back to Fig. 2, an optimal product offering will be that product which serves the

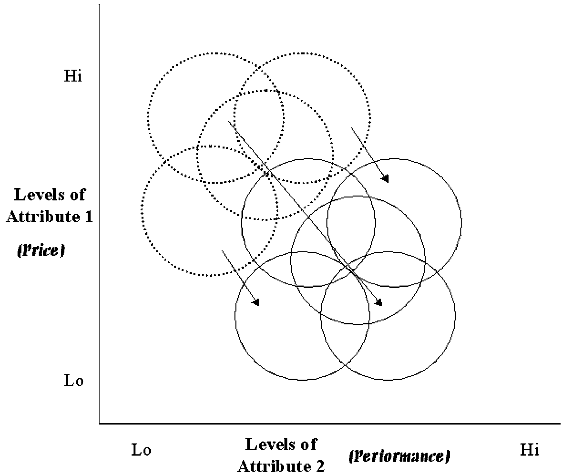

Fig. 2. Perceptual map with shifting consumer preference over time.

largest number of customers over time on the two dimensions of price and performance. In theory, this point will exist on the map where the maximum number of customer preference circles overlap.

2.4. Ant colony optimization

Ant colony optimization (ACO) techniques model the social behavior of ants in an attempt to solve combinatorial optimization problems. ACO techniques are part of a larger field of study called swarm intelligence (SI). Bonabeau et al. (1999, 2002) present the first interdisciplinary approach to the field of SI, showing that there are many applications where social insect behavior can be used as reference for modeling purposes. In particular, these authors emphasize the tremendous potential of SI for combinatorial optimization problems. Originally, SI was applied to robotic systems and artificial life, where simple rules of conduct and models were patterned from insect behavior. Now, SI is applied to a variety of disciplines, including business applications. Ant colony behaviors have been modeled commercially in companies such as Southwest Airlines where ant-foraging behavior is simulated for efficient vehicle routing at tremendous cost savings, an unnamed major retail chain where ant seed-harvesting and food transferring behaviors are mimicked for more efficient order picking strategies (Bonabeau and Meyer, 2001), and at Unilever where ant-based methods have been used to decrease the time to complete a variety of tasks at one of their large production facilities (Bonabeau and Theraulaz, 2000).

A major application for ACO comes from observing how ants forage for food. When foraging for food, individual ants, as simple creatures, follow a series of local rules, such as: (1) following those paths that have been frequented by fellow member ants or, specifically, following paths that have high levels of chemical pheromone on its trail, and (2) laying pheromone on own trail to communicate with future ants who might follow. If each individual ant follows these simple rules, a sophisticated form of colony decision-making emerges from a society of simple rule followers (Dorigo et al., 1999).

Over time, as ants forage for food, positive reinforcement, in the form of pheromone scent from previous ants, leads new ants to a food source via more direct routes. Fig. 3 represents how a pheromone trail works to "organize" ants toward better solutions over time.

Bonabeau and Meyer (2001), state that social insects are an evolutionarily successful class of insects due to flexibility, robustness, and the ability to self-organize. Social insects are flexible and can adapt quickly to changing environmental conditions. They are robust in that if individual members of the group fail at given

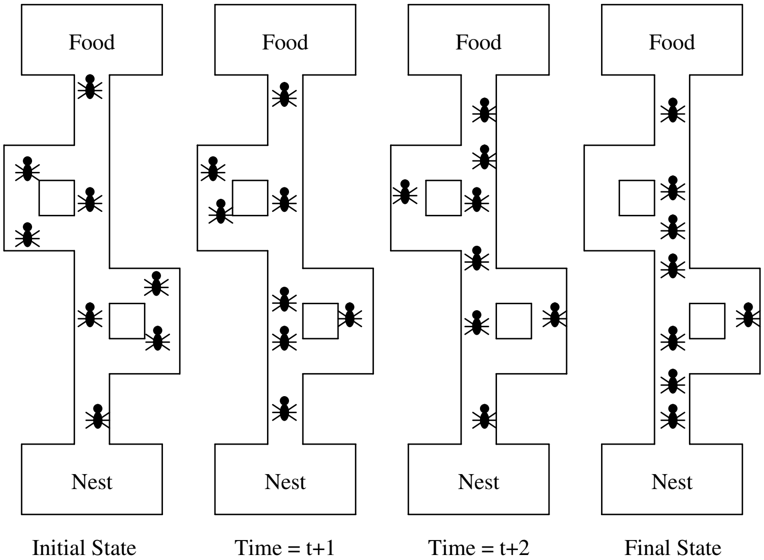

Fig. 3. Dynamics of ant-generated pheromone trail over time. *Source*: McMullen et al. (2001).

tasks, the group can still be successful. They self-organize into colony decision-makers using local information. These qualities of flexibility, robustness and self-organization make ACO a viable search heuristic for problems designed around changing or uncertain conditions. How ACO algorithms are patterned after social ant foraging techniques is described below.

At initialization, time period 0, ants will leave the nest in a random pattern to forage for food. These ants possess only a local perspective to guide them towards food sources. As ants leave the nest, they deposit small amounts of pheromone on their trail. This chemical is a message to future ants that the path has previously been taken for food.

Ants have a predisposition to follow pheromone. The larger the amount of pheromone, the more likely from a probabilistic standpoint an ant is to follow this trail. This chemical message is central to the success of the ant colony. It is useful because ants who make the best decisions will return to the nest first, leaving their trail with twice the amount of pheromone than another path taken by another ant who has left for food and has yet to return. Over time, this quality self-organizes ants toward efficient food foraging solutions.

At time period $t + 1$, the next wave of ants will go out seeking food. At this time, many of the initial set of foraging ants have set out for and returned from food expeditions. The new generation of ants makes decisions on which path to take toward an unknown food source based on simple rules of engagement. A new ant will make a decision whether or not to take a specific route based on the amount of pheromone that exists on that path. This decision follows a probabilistic distribution; the more pheromone on a path, the higher the likelihood a new ant will take that path. Conversely, the less the pheromone, the less likely an ant is to follow this path. Over time (time periods $t + 2, \ldots, t + n$), the iterative process of simple ants following paths left by their predecessors, leads to the emergence of an efficient path toward the food source.

Because ants follow a probabilistic distribution for path selection there is still a probability, albeit small, that an ant will take an "undiscovered" path. This attempts to prevent convergence upon local optima.

Ant optimization techniques offer virtual ants enhancements over natural ants. With enhancement, virtual ants can make "better" decisions about which paths will lead them to food sources because these ants can take advantage of not only local information, but also global information, and a history of decisions. Enhancements such as pheromone evaporation, for example, modify the probabilistic distribution of subsequent ant decisions. With evaporation, pheromone levels on paths not recently taken begin to diminish.

With lower levels of pheromone, the probability of an ant taking this path is minimized. With evaporation, pheromone left by previous generation ants will slowly evaporate, so that new ants will not converge on local optima and are able to adapt to changing market conditions (e.g. existing food source is depleted, or new sources of food are discovered).

Dorigo et al. (1999) organized a variety of ant colony behavioral algorithms into a common framework and presented a novel meta-heuristic named ACO. With this heuristic, a simulated colony of individual ants move through the problem space by applying local stochastic rule policy. As they move through the problem space, the ants incrementally build solutions to the problem, evaluating the quality of their partial-step solution, and depositing information on their path in the form of a pheromone message that directs future ants toward iteratively better solutions. Please note that an individual ant's decision is not being treated as analogous to a consumer's decision. An individual ant's decision process is not sophisticated enough to serve as a proxy for human decisions. We are merely using a colony of virtual ants to iteratively help find the best mix of product characteristics based upon an objective function.

3. Methodology

The symbols and definitions for each variable in the PD problem and ACO heuristic are listed in Table 1.

3.1. Detailed description of the research modeling

Fig. 4 generalizes the overall process for how the PD selection approach is modeled. Each step of the problem is outlined through initialization, search, probabilistic updates, and multiple iterations. First, the ant colony optimization heuristic approach is described in detail.

3.2. A matrices

The first step involves the initialization of the problem, starting with the part-worths matrices. The part-worths matrices in a conjoint analysis contain all the actual consumer preference information for every level of each attribute. Here, however, the lower and upper bounds for these matrices are user-specified. These values are then used to compute the a_{jk} values for each of the **A** matrices, for each period p, and are determined via the following relationship:

$$a_{jk}^{(p)} = \mathrm{LB} + \mathrm{rand} * (\mathrm{UB} - \mathrm{LB}). \tag{1}$$

The value of rand is a uniformly distributed random number on the [0, 1] interval. This matrix is created for each period (Periods times), and each $\mathbf{A}^{(p)}$ matrix has Cust rows and Att $*$ Levels columns. For each subsequent period, consumer preferences, or a_{jk}, will evolve, via a diffusion process. The inclusion of this diffusion process is unique to the PD problem. In the past, consumer preferences have been treated as unchanging variables.

The following formula represents this mathematically:

$$a_{jk}^{(p)} = a_{jk}^{(p-1)} - \delta + \varepsilon. \tag{2}$$

3.3. G matrices

The history matrix is initialized for all values of g_{jk}. The **G** matrix tracks the history of best path solutions. The **G** matrix is defined as Att rows and Levels columns. To initialize the **G** matrix, all values of g_{jk} are set to 1.

Table 1
Variables and definitions

Symbol	Definition
Ants	Number of ants used for search
Cust	Number of customers
Att	Number of unique product attributes
Levels	Unique levels for each attributes
Periods	Number of time periods
i	Cost of customers
j	Attribute for a product offering
k	Level of a given attribute for a product offering
LB	Lower bound for part-worths matrix
UB	Upper bound for part-worths matrix
$\mathbf{A}^{(p)}$	Part-worths matrix of interest for period p
$a_{jk}^{(p)}$	Specific value of $\mathbf{A}$, for level k for given attribute j, for period p
Captured_{jk}	Customers captured for period when selecting level k for attribute j
Prob_{jk}	Probability of selecting level k for attribute j
NPVProb_{jk}	Probability of selecting level k for attribute j using NPV criterion
LocProb_{jk}	Local probability of selecting level k for attribute j
GlobProb_{jk}	Global probability of selecting level k for attribute j
γ	NPV update factor (weight)
α	Local update factor (weight)
β	Global update factor (weight)
η	Local update adjustment factor
Z	Objective function value
Z^*	Optimal objective function value obtained via search
Z^{**}	Objective function value obtained via complete enumeration
$\mathbf{G}$	History matrix (an Att × Levels matrix)
g_{jk}	History matrix component for level k of attribute j
δ	Period decrease value
ε	Uniformly distributed random value on the $[-0.5, 0.5]$ interval
Percentile	Percentile performance
Inferiority	Inferiority percentage
Superiors	Solutions found via complete enumeration superior to Z^*

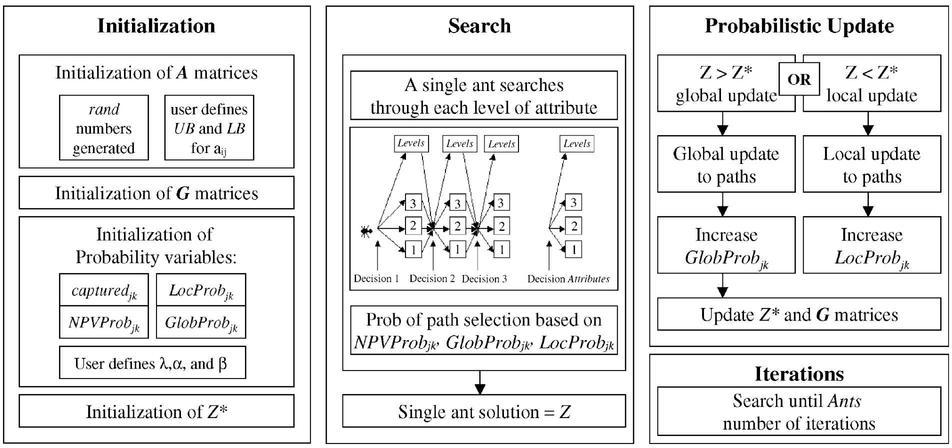

Fig. 4. Methodology for ant colony optimization search.

3.4. NPV probabilities

To determine the NPV probabilities, the number of customers captured when selecting level k of attribute j must first be determined. Captured_{jk} is an important component as it tracks the number of consumers who will be satisfied by a particular product offering. Obviously, good solutions will have a large "captured" audience. The value of Captured_{jk} is initialized to zero. The Captured value for each level k of attribute j will remain constant, or increase, via the properties of Eqs. (3) and (4), respectively. The following logic is used mathematically:

$$\text{Captured}_{jk} = \text{Captured}_{jk} \quad \text{if} \sum_{p=1}^{\text{Periods}} a_{jk}^{(p)} \leqslant 0, \tag{3}$$

$$\text{Captured}_{jk} = 1 + \text{Captured}_{jk} \quad \text{if} \sum_{h=1}^{\text{Periods}} a_{jk}^{(p)} > 0. \tag{4}$$

The value of NPVProb_{jk} represents the relative desirability of selecting level k for attribute j for all Periods of the assumed product life. It is important to note that the NPVProb_{jk} values remain constant throughout the ACO search.

Mathematically, this is expressed as follows:

$$\text{NPVProb}_{jk} = \frac{\sum_{h=1}^{\text{Periods}} \text{Captured}_{jk} \cdot (1+i)^{-h}}{\sum_{k=1}^{\text{Levels}} \sum_{h=1}^{\text{Periods}} \text{Captured}_{jk} \cdot (1+i)^{-h}}. \tag{5}$$

The local and global probabilities of selecting level k for attribute j are initialized to the NPVProb_{jk} values. Unlike the NPVProb_{jk} values, the LocProb_{jk} and GlobProb_{jk} values will change as solutions are attained. Mathematically, the initialization for LocProb_{jk} and GlobProb_{jk} are expressed as

$$\text{LocProb}_{jk} = \text{NPVProb}_{jk}, \tag{6}$$

$$\text{GlobProb}_{jk} = \text{NPVProb}_{jk}. \tag{7}$$

The optimal objective function value, Z^*, will be used to track the best decisions. Z^* is initialized to zero.

3.5. Search

After initialization, the search process begins. As stated, the search process for a single ant is based on local information. An ant makes "decisions" based on a structure similar to Fig. 5. Here, an ant will go through several decision levels (three, as shown, in the diagram) based on the "desirability" of each attribute characteristic (as shown, levels 1, 2 or 3). To reiterate, "desirability" is based on a probability distribution containing attribute desirability, or NPVProb_{jk}, and the amount of pheromone (LocProb_{jk} and GlobProb_{jk}) on the trail. Each time an ant is faced with a decision regarding the choice of a certain level of attribute, a probability of selection is constructed. An ant will move through the system from "decision" to "decision", from each level to the next level, until the ant reaches the final attribute and a solution to the problem is found. Mathematically, the probability of an ant choosing level k of attribute j is as follows:

$$\text{Prob}_{jk} = \frac{\gamma \cdot \text{NPVProb}_{jk} + \alpha \cdot \text{LocProb}_{jk} + \beta \cdot \text{GlobProb}_{jk}}{\sum_{k=1}^{\text{Levels}} \gamma \cdot \text{NPVProb}_{jk} + \alpha \cdot \text{LocProb}_{jk} + \beta \cdot \text{GlobProb}_{jk}}, \tag{8}$$

where

$$\gamma + \alpha + \beta = 1. \tag{9}$$

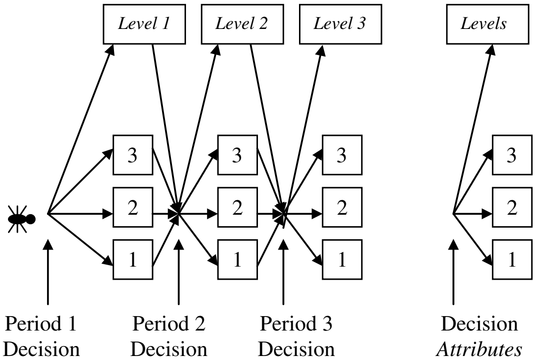

Fig. 5. A single ant's journey through the decision network. *Source*: McMullen et al. (2001).

In this formulation, γ, α, and β are user-specified parameters, indicating the weights assigned to NPVProb_{jk}, LocProb_{jk} and GlobProb_{jk}, respectively. These parameters affect the degree of priority given to original consumer preference and all local and global updates to a given path_{jk}. The parameters: λ, α, and β sum to 1, mathematically, as shown in Eq. (9).

This probabilistic selection approach continues through each of the j attributes of the problem. When the ant completes its "journey", the problem is considered solved, and the objective function value is determined for that particular solution. Mathematically, the objective function, Z, is represented as follows:

$$Z = \sum_{h=1}^{\text{Periods}} \text{Captured}_{jk} \cdot (1+i)^{-h}, \quad \text{where } j,k \in \text{solution}. \tag{10}$$

It is emphasized here that the value Captured_{jk} is the combination of attribute selections for the ant of interest's journey. Please refer to Eqs. (3) and (4).

3.6. Probabilistic updating

The value of Z represents one ant's "journey", or one ant's solution to the problem. This is not necessarily the best solution. The objective function value, Z, is compared against the optimal objective function value, Z^*. If the objective function value is an improvement over the optimal, a global update procedure is performed; otherwise, a local update procedure is performed. Global updates model the biological process whereby an ant lays a pheromone trail on the path it has just taken. Local updates represent pheromone evaporation. As new ants come across "inferior" solutions, the amount of pheromone on that path "evaporates" to lessen the probability of a new ant making the same mistake. Mathematically, this is as follows:

$$\text{If } Z > Z^*, \quad \text{global update,} \tag{11}$$

$$\text{If } Z \leqslant Z^*, \quad \text{local update.} \tag{12}$$

The local update is performed to encourage exploration of "new" solutions—subsequently reducing the probability of future ants selecting the most recently chosen solution and moving along the same path as that which was just selected. As discussed, local updates will occur along the paths of inferior solutions. Mathematically, the local update is as follows (the value of η is user-chosen):

$$\text{LocProb}_{jk} = \text{LocProb}_{jk} \cdot (1-\eta). \tag{13}$$

Conversely, the global update is performed to enhance the probability of selecting the path components of the recently found optimal solution. First, the history matrix, **G**, is updated to reflect the recently found optimal solution according to the following relationship:

$$g_{jk} = g_{jk} + 1, \quad \text{where } j, k \in \text{optimal solution}. \tag{14}$$

Second, the GlobProb$_{jk}$ values are updated to account for the new values of g_{jk}. This is done according to the following relationship:

$$\text{GlobProb}_{jk} = \frac{g_{jk}}{\sum_{h=1}^{\text{Levels}} g_{jk}}, \quad \text{where } j, k \in \text{optimal solution}. \tag{15}$$

After the completion of the necessary local and global updating, the probability of selecting level k for attribute j is readjusted using Eq. (8).

3.7. Iterations

Following initialization, the search procedure will continue for *Ants* number of simulated ants, or iterations. Ants is a user specified parameter. At the completion of the search, the best solution, or the final Z^* value, is found and its associated details are reported to the user. An example problem is included as an Appendix A.

4. Experimentation

4.1. Test problems

Four test problems of varying size and complexity are used for assessment. The details of these problems are below in Table 2.

The number of feasible solutions is determined by the following relation, which strongly suggests the combinatorial difficulty of these problems:

$$\text{Feasible solutions} = \text{Levels}^{\text{Att}}. \tag{16}$$

It should also be noted that the number of ants used in the search (Ants) is also given above, and this value is proportional to the number of feasible solutions.

Table 2
Test problems used for assessment

Problem	Attributes (Att)	Levels (Levels)	Customers (Cust)	Periods (Periods)	Ants (Ants)	Feasible solutions
1	12	5	300	10	1,464,844	244,140,625
2	5	10	200	5	600	100,000
3	8	10	250	5	600,000	100,000,000
4	9	10	250	7	6,000,000	1,000,000,000

4.2. Search parameters and research questions

It is desired to determine the efficiency of the search heuristic described—how well an ACO solution compares to the true optimal combination of user-chosen product attributes. In order to investigate this question, it is important to select user-controlled parameter values that provide the most desirable solutions (again, the most desirable solution has the highest value of Z^*). These user-controlled parameter values are the aggressiveness of local probability updating (η), as well as the three weights for NPV, global and local probabilities (γ, β and α, respectively). From this, the following research questions are posed:

1. Does the aggressiveness of local probability updating have an effect on the objective function value? If so, what values of η are most beneficial?
2. Does the combination of probability weighting have an effect on the objective function value? If so, what combination is most beneficial?
3. Is there a general relationship between the value of the probability weighting value and the performance of the search heuristic?

If the research questions above reveal any significant findings, the values of the user-chosen parameters resulting in a maximized objective function will be used throughout the remainder of the paper.

The first research question is addressed by using two different values of local updating ($\eta = 0.05$ and $\eta = 0.10$). The second research question is addressed by using different values of the probability weighting factors of γ, β and α. These values are varied in increments of 0.2, with the following possible combinations shown in Table 3.

For each unique combination of parameters, the search was performed 25 times. This results in a full-factorial experimental design consisting of 4200 observations:

$$(4 \text{ problems})(2 \text{ values of } \eta)(21 \text{ weight combinations})(25 \text{ replications}) = 4200.$$

Table 3
Possible combinations of probability weighting factors

Comb.	γ	α	β
1	0.0	0.0	1.0
2	0.0	0.2	0.8
3	0.0	0.4	0.6
4	0.0	0.6	0.4
5	0.0	0.8	0.2
6	0.0	1.0	0.0
7	0.2	0.0	0.8
8	0.2	0.2	0.6
9	0.2	0.4	0.4
10	0.2	0.6	0.2
11	0.2	0.8	0.0
12	0.4	0.0	0.6
13	0.4	0.2	0.4
14	0.4	0.4	0.2
15	0.4	0.6	0.0
16	0.6	0.0	0.4
17	0.6	0.2	0.2
18	0.6	0.4	0.0
19	0.8	0.0	0.2
20	0.8	0.2	0.0
21	1.0	0.0	0.0

Assuming that the second research question uncovers that weighting has a significant effect on the objective function, it is desired to compare the objective function values resultant from the "best" combination of weighting factors with the optimal values of the objective function obtained via complete enumeration.

Because four different problems are used, radically differing objective function values will result. To compensate for this, and to permit generalized analyses *across* the four problems (and for *all* research questions), objective function values are compared against the true optimal objective function value (obtained via complete enumeration) in two different forms. The first form is percentile performance (Percentile), the second form is inferiority percentage (Inferiority). Percentile performance is computed as follows:

$$\text{Percentile} = 100 * (1 - (\text{Superiors}/(\text{Levels}^{\text{Att}}))). \tag{17}$$

Here, Superiors is the number of solutions found via complete enumeration having objective function values superior to that of Z^{*}. The computation for Inferiority is a direct comparison between the objective function value found via search (Z^{*}) and that found via complete enumeration (Z^{**}).

$$\text{Inferiority} = 100 * (1 - (Z^{*}/Z^{**})). \tag{18}$$

From inspection of Eqs. (17) and (18), we hope to have a Percentile value of 100% and an Inferiority value of 0%—or as close as possible to these values.

With detail of objective function values presented, it is desired to then determine whether or not the solutions obtained from the "best" combination of user-defined parameters are statistically different from the absolute optimal values gathered via complete enumeration.

This is formally stated with the fourth and final research questions:

4. In terms of percentile performance, does the presented methodology provide results that are statistically no different from true optimal solutions? Mathematically, this research question is presented via the following set of hypotheses:

$$H_0: \quad \mu_{\text{Percentile}} = 100,$$
$$H_A: \quad \mu_{\text{Percentile}} < 100.$$

5. In terms of inferiority performance, does the presented methodology provide results that are statistically no different from the true optimal solutions? Mathematically, this research question is presented via the following set of hypotheses:

$$H_0: \quad \mu_{\text{Inferiority}} = 0,$$
$$H_A: \quad \mu_{\text{Inferiority}} > 0.$$

As supported by Neter et al. (1996), univariate ANOVA is used to address the first two research questions, simple linear regression is used for the third research question, while *t*-tests are used to address the fourth and fifth research questions.

4.3. Computational experience

The C++ programming language was used to construct both the ant-colony search heuristic and the enumeration program. The operating system used was Microsoft Windows XP; the processor was an Intel Pentium IV, with a clock speed of 2.26 GHz.

4.4. Experimental results and observations

The first research question tests if the local probability-update factor has an effect on performance. We found no statistical support for local updates effect upon better solutions ($F = 0.077$, $p = 0.781$ for Percentile; $F = 0.706$, $p = 0.401$ for Inferiority). Table 4 details Percentile and Inferiority Performance organized by value of η. No statistically meaningful performance differences in terms of η are noticed, questioning the use of pheromone evaporation.

The second research question explores the effect that the weighting combination has on performance. The weighting combination is found to have an effect on performance ($F = 12.779$, $p < 0.001$ for Percentile; $F = 82.258$, $p < 0.001$ for Inferiority). Table 5 shows a breakdown of performance by each unique weighting combination.

The first weighting combination is highlighted as it provides the most desirable levels of both Percentile and Inferiority. This weighting combination is quite interesting, in that this combination includes only global probability updating ($\beta = 1$). NPV probability weighting and local probability weighting are not represented ($\gamma = \alpha = 0$), as shown in the first row of Table 11.

Table 6 provides information required to respond to the third research question. This table shows that as the NPV and local weighting factors increase, solution quality deteriorates—Percentile performance decreases and Inferiority increases. Global weighting has the opposite effect—as its value increases, Percentile

Table 4
Percentile and inferiority performance by local probability update factor: means (standard deviations)

η value	Percentile	Inferiority
0.05	99.9987 (0.0029)	0.1808 (0.2956)
0.10	99.9989 (0.0023)	0.1721 (0.2847)

Table 5
Percentile and inferiority performance by weighting combination: means (standard deviations)

Comb.	Percentile	Inferiority
1	**99.9988 (0.0026)**	**0.1764 (0.2895)**
2	99.9974 (0.0105)	0.2247 (0.3272)
3	99.9944 (0.0175)	0.2006 (0.4102)
4	99.9962 (0.0118)	0.2284 (0.3726)
5	99.9977 (0.0064)	0.2057 (0.3546)
6	99.9499 (0.1158)	1.1830 (0.6802)
7	99.9849 (0.0388)	0.3228 (0.5855)
8	99.9939 (0.0234)	0.2645 (0.4387)
9	99.9934 (0.0167)	0.2805 (0.4459)
10	99.9822 (0.0519)	0.3876 (0.6050)
11	99.9670 (0.0926)	0.9854 (0.6283)
12	99.9939 (0.0159)	0.3120 (0.4588)
13	99.9838 (0.0496)	0.3754 (0.5833)
14	99.9732 (0.0672)	0.5351 (0.6764)
15	99.9508 (0.1300)	1.0631 (0.6519)
16	99.9886 (0.0287)	0.3815 (0.5414)
17	99.9814 (0.0453)	0.5286 (0.5984)
18	99.9707 (0.0724)	0.9688 (0.6174)
19	99.9705 (0.0723)	0.6794 (0.6334)
20	99.9614 (0.0847)	1.0384 (0.6224)
21	99.9601 (0.0644)	1.0286 (0.6462)

Table 6
Regression results for weighting factors

Weighting factor	Percentile slope (t-stat; p-value)	Inferiority slope (t-stat; p-value)
γ (NPV)	−0.026 ($t = -7.844$; $p < 0.001$)	0.591 ($t = 18.46$; $p < 0.001$)
α (Local)	−0.015 ($t = -4.509$; $p < 0.001$)	0.339 ($t = 10.33$; $p < 0.001$)
β (Global)	0.041 ($t = 12.512$; $p < 0.001$)	−0.930 ($t = -30.988$; $p < 0.001$)

performance increases, with Inferiority performance decreasing (see highlighted row). This particular discovery is in concert with the findings associated with the second research question—heavier weighting of global probability or positive reinforcement results in more desirable solutions in terms of the objective function.

Given that the first weighting combination ($\beta = 1, \gamma = \alpha = 0$) results in the most desirable solutions, this weighting combination alone will be used to address Research Questions 4 and 5. The percentile of solutions for this specific weighting combination that the ACO approach surpasses, or Percentile, is 99.9988, indicating strong performance of the heuristic. Since we are comparing performance to optimal using our four problems, we are seeking a value of 100% to state true optimality. Here, the mean Percentile performance of the search heuristic does not provide a value statistically the same as 100% ($t = -6.25$, $p < 0.001$). Looking to the other means of measuring optimal performance, the mean Inferiority performance does not provide a value statistically the same as 0%, the desired value to state optimality ($t = 8.618$, $p < 0.001$). Although Research Questions 4 and 5 do not provide results that permit the decision-maker to claim statistically optimal performance of the search heuristic, the authors do take comfort in the results being *near*-optimal.

5. Discussion and summary

5.1. Significant findings

Ant colony algorithms generate near optimal results when applied to the PD problem. High quality heuristic procedures are often essential for the modeling of practical large-scale applications. Existing PD formulations which are designed to find true optimality are typically overly-constrained to make the problem manageable from a computational standpoint. While this study does not compare ACO algorithms to other heuristic procedures such as genetic algorithms, ACO appears to be a highly practical heuristic approach for this particular problem.

In addition to presenting a new heuristic approach to the PD problem, another key contribution is the use of changing consumer preference levels over time. Prior studies have been overly constrained in that they treat consumer inputs as static. By modeling consumer preferences as shifting over time, product designs are more robust, and hence, less risky in global business environments where consumers are increasingly more demanding.

A tertiary contribution of this study is how the use of pheromone signaling affects the quality of ant-generated solutions. Prior to this research, the authors have found no empirical research which tests individual behaviors such as global updating (positive reinforcement using pheromone deposit) or local updating (evaporation of pheromone over time). Of significant importance to the SI and ACO literature, global updating and weighting combination were found to be the key contributing factors in generating high quality solutions to the PD problem. It seems logical that the use of pheromone deposit will be a key component as it is a primary tool used by social insects to communicate with each other and to "solve" problems. Perhaps global reinforcement became so critical for this particular endeavor because of the way we modeled the PD problem. It seems reasonable that local updates would become more critical in environments with extreme amounts of change (analogous to an ant colony searching for food sources that are constantly depleting and being found in new areas). Consequently, we do not recommend that future formulations omit local updating altogether. Quite the opposite, we hope future work will test individual ant behaviors under different market conditions to gain a better assessment of what factors contribute to the success of ACO.

5.2. Discussion

When selecting a heuristic for an optimization problem, the needs of the specific problem formulation should be addressed. For formulations where flexibility is essential due to changing market conditions or general uncertainty, ACO is a highly viable alternative. ACO mirrors social ant behavior, an evolutionarily successful class of insects. These insects are successful because of their flexibility, robustness and the ability to self-organize. These qualities make for a particularly attractive heuristic procedure. ACO heuristics are especially viable for the PD problem because simulated ants can handle discrete product preferences like color or size, as well as continuous preferences like degrees of "redness" or degrees of size. Simulated ants can also handle categorical qualitative preferences; of course translating these qualitative preferences into product attributes could potentially present a problem when incorporating these plans into workable product designs.

The PD problem has a relatively long, well-developed history. Because of this history, whenever considering a new approach to this problem, a core question should be asked: What factors should one look for when choosing a new approach to solving this problem? Our approach in developing this problem was to interject a level of complexity that allows for heightened realism. Uncertainty and complexity are of considerable import to PD problem modelers as future consumer desires for product characteristics cannot be definitively judged, and as product designing becomes increasingly more complex. Unfortunately, making product design models realistic is often very difficult to realize. At a minimum, the desired outcome should be better decision making. In this vane, we deemed it critical to present an approach that was both flexible to environmental changes and robust enough to handle large amounts of input data.

In the interest of better decision models, we framed the PD problem using a model of consumer decision making. A significant benefit of this theoretical approach is that consumer preference information can become a critical input for a decision support system designed to maximize firm profits. Such an approach allows for scenario planning where changes in product characteristics can be evaluated to forecast the potential impact on market share, revenues, and firm profit.

Additionally, because conjoint data can be used for tangible design decisions, the value of conjoint data is vastly improved. Conjoint data is generally considered to be highly useful in understanding consumers' relationships with existing and prototype products. And while conjoint data techniques are the norm for gathering this type of data, they are often criticized because they are difficult to administer, often cost prohibitive and frequently very time-consuming.

5.3. Limitations and directions for future research

A main limitation of the current study is the use of a simulated part worth matrix of consumer preferences. While simulated data is essential for statistical control and necessary to test the presented hypotheses, it is a limiting factor when we try to generalize findings to a real world application. There is an opportunity to strengthen the current study by extending this methodology to PD problems where actual consumer preference data is modeled into a DSS for product positioning and design.

Additionally, the current study is limited in that it just makes a comparison between ACO-generated solutions and the complete enumeration of all possible solutions. While ACO solutions were found to be near-optimal, future research on optimal PD should address ACO algorithms as compared to other heuristic techniques, such as genetic algorithms, simulated annealing, beam search and tabu search, to name a few. This is a necessary next step to ensure that ACO is, indeed, an appropriate heuristic for PD. ACO algorithms have been successfully compared to other heuristics for different large-scale combinatorial optimization problems with extremely promising results; these problems include the traveling salesman problem (Dorigo et al., 1996), job shop scheduling (Colorni et al., 1994), telecommunications network routing (DiCaro and Dorigo, 1998), and sequential ordering (Gambardella and Dorigo, 2000). The intent of the current study is not to provide a comparison of heuristics, especially given that the ACO approach is compared to optimal solutions. The impetus here is to address this complex problem using a new and promising approach. Comparisons to other heuristics are left as an opportunity for future research.

Future research should also consider extending ACO algorithms from the single PD problem to the more complex product line problem, where cannibalization of existing products in the line must be considered in concert with engineering costs and customer requirements. The current study establishes the groundwork for this endeavor, as the presented theoretical model and ACO methodology can be easily applied to multiple products in a line.

Lastly, the current study is successful at highlighting those components of ant colony optimization (e.g., initial preference weighting, local updates, and global updates) that are the most robust in terms of generating good solutions for design problems of varying size and complexity. While we feel these results can be generalized to other problems, the validity of these results should be empirically tested. It is strongly encouraged that similar studies of individual ACO behaviors should be made on other large-scale combinatorial optimization problems, such as the sequential ordering and network routing problems.

Appendix A. Example problem

To further illustrate how ACO works, please consider a simple product design with two attributes (j), each having three levels of attribute (k), over two periods of time. Ten customer inputs or preferences are assumed and presented. This problem is presented through all model steps: initialization, search, and iteration.

To initialize this problem, we first construct the $\mathbf{A}^{(p)}$ matrices. For the first time period, Eq. (1) is employed using a random number generation scheme via the user-defined parameters of $\mathrm{LB} = -5$ and $\mathrm{UB} = 5$. Information for the second time period is generated by transforming this information via a user-defined diffusion factor, $\delta = 0.1$, and adding in a random term, ε on the $[-0.5, 0.5]$ interval. Table 7 shows the part-worths, or $\mathbf{A}^{(p)}$ matrix for the problem. For every customer, the preference for a given level of attribute (e.g., L1, L2, or L3) is noted for each of the two time periods. If the value shown in the matrix for a given product attribute level is positive, a customer will react favorably to that attribute. Conversely, it is assumed that a customer will react negatively if that integer is negative.

Next, we initialize the $\mathbf{G}$ matrix, or the history matrix, to 1. Here the rows represent the attribute and the column represents the level of attribute. We initialize the history matrix to 1 to ensure that every level has an equal probability in global updating. Initializing the matrix to zero would force zero probability of global update for any level. $\mathbf{G}$ matrix $= \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$, attribute rows by level columns. The next step is to calculate the NPV probabilities ($\mathrm{NPVProb}_{jk}$) via Eqs. (3)–(5). Table 8 shows the number of customers "captured" for each level of each attribute. As discussed, a customer is counted as captured if there is a positive response (e.g., positive integer value) for the corresponding level of the $\mathbf{A}^{(p)}$ matrix.

Next, a NPV matrix based on these capture levels is computed. Future captures are discounted via a user defined discount rate of i cost of capital. For this example, a discount rate of 5% was used to discount future

Table 7
Sample problem **A** matrix

Customer #	Time period 1						Time period 2					
	Attribute 1			Attribute 2			Attribute 1			Attribute 2		
	L1	L2	L3	L1	L2	L3	L1	L2	L3	L1	L2	L3
1	−4.40	1.27	3.90	4.41	−4.83	−3.08	−5.45	0.26	3.04	3.54	−5.96	−3.85
2	2.77	−0.73	−1.46	4.10	−0.32	2.66	1.68	−1.92	−2.48	3.20	−1.08	1.44
3	0.54	2.58	0.21	−3.85	−4.16	−3.67	−0.59	1.58	−0.75	−4.92	−5.10	−4.84
4	−2.71	1.44	4.24	−4.82	−4.48	3.05	−3.85	0.60	3.27	−5.69	−5.60	2.24
5	−4.45	3.16	4.72	−2.40	−1.57	−1.95	−5.40	2.22	3.73	−3.22	−2.51	−2.94
6	−4.48	−3.10	2.38	−4.11	−1.37	−1.79	−5.61	−3.91	1.55	−5.24	−2.62	−2.97
7	−2.72	4.11	−3.31	0.62	−2.71	0.71	−3.87	3.35	−4.49	−0.53	−3.65	−0.04
8	0.41	3.77	−4.79	−4.00	3.99	−1.74	−0.80	2.99	−5.56	−5.08	3.05	−2.97
9	−0.15	4.61	−2.41	4.36	4.85	4.97	−1.13	3.52	−3.55	3.60	3.74	3.81
10	−3.41	3.85	2.93	3.79	1.55	3.58	−4.17	2.80	1.95	2.94	0.65	2.61

Table 8
Sample problem customer "Capture" matrix

Customer #	Time period 1						Time period 2					
	Attribute 1			Attribute 2			Attribute 1			Attribute 2		
	L1	L2	L3	L1	L2	L3	L1	L2	L3	L1	L2	L3
1	0	1	1	1	0	0	0	1	1	1	0	0
2	1	0	0	1	0	1	1	0	0	1	0	1
3	1	1	1	0	0	0	0	1	0	0	0	0
4	0	1	1	0	0	1	0	1	1	0	0	1
5	0	1	1	0	0	0	0	1	1	0	0	0
6	0	0	1	0	0	0	0	0	1	0	0	0
7	0	1	0	1	0	1	0	1	0	0	0	0
8	1	1	0	0	1	0	0	1	0	0	1	0
9	0	1	0	1	1	1	0	1	0	1	1	1
10	0	1	1	1	1	1	0	1	1	1	1	1
Sum	3	8	6	5	3	5	1	8	5	4	3	4

captures. Table 9 details this matrix, which was calculated by taking the total number of customer captures by attribute level by year two and discounting these values to time period zero. Here, 1.86 represents two future captures discounted 5% for two years, with 0.95 representing a one future capture discounted at the same rate. Please note future captures will either be values of 0, 1 or 2, corresponding to a discounted value of 0, 0.95, or 1.86.

From these values, the NPVProb_{jk} is calculated via Eq. (5), as shown in Table 10. For illustration purposes, let's define attribute 1 as color and attribute 2 as size. Here, if an ant were exclusively using NPVProb_{jk} criteria to base its decisions, it would make decisions based on the following probability distribution. For its first level of decisions (attribute 1), it would select the color red 13.03% of the time, the color blue 51.49% and the color green 35.48%. For its second level of decisions (attribute 2), it would select the size small 37.53% of the time, medium 24.95% and large 37.53% of the time. Please note that these probability values are set and will not change during the entire ACO search procedure.

Finally, LocProb_{jk} and GlobProb_{jk} values are initialized to NPVProb_{jk} levels using Eqs. (6) and (7). Unlike NPVProb_{jk}, these values will change over time. These values will be updated and modified as the search procedure goes through its many iterations.

Table 9
Sample problem NPV matrix

Customer	Attribute 1			Attribute 2		
	L1	L2	L3	L1	L2	L3
1	\$0.00	\$1.86	\$1.86	\$1.86	\$0.00	\$0.00
2	\$1.86	\$0.00	\$0.00	\$1.86	\$0.00	\$1.86
3	\$0.95	\$1.86	\$0.95	\$0.00	\$0.00	\$0.00
4	\$0.00	\$1.86	\$1.86	\$0.00	\$0.00	\$1.86
5	\$0.00	\$1.86	\$1.86	\$0.00	\$0.00	\$0.00
6	\$0.00	\$0.00	\$1.86	\$0.00	\$0.00	\$0.00
7	\$0.00	\$1.86	\$0.00	\$0.95	\$0.00	\$0.95
8	\$0.95	\$1.86	\$0.00	\$0.00	\$1.86	\$0.00
9	\$0.00	\$1.86	\$0.00	\$1.86	\$1.86	\$1.86
10	\$0.00	\$1.86	\$1.86	\$1.86	\$1.86	\$1.86
Sum	\$3.76	\$14.88	\$10.25	\$8.39	\$5.58	\$8.39

Table 10
Sample problem NPV probabilities

	Color (Attribute 1)			Size (Attribute 2)		
	Red (L1)	Blue (L2)	Green (L3)	Small (L1)	Medium (L2)	Large (L3)
NPV sum	\$3.76	\$14.88	\$10.25	\$8.39	\$5.58	\$8.39
Probability (%)	13.03	51.49	35.48	37.53	24.95	37.53

The objective function for the search procedure is Z^* (initialized to zero at start of the search), set to find a combination of product attributes that will maximize the number of customers over time that will choose that product. Each simulated ant will move through decision space making decisions, and will report a final solution, or Z value. After each search iteration, the reported Z value is compared to the global Z^*, to ascertain whether an improvement has been made. A user-defined number of iterations, Ants, are completed. It should be noted that this sample problem illustrates only two Ants so an "optimal" solution is not expected.

The search procedure for a large problem is completed via Monte Carlo simulation. For this simple problem, a uniform random number in the interval $[0, 1]$ for each attribute decision is generated. This value is then compared to the cumulative probabilities shown in Table 11. These cumulative probabilities will work for the first iteration because LocProb_{jk} and GlobProb_{jk} have each been initialized to the NPVProb_{jk} value. The random number lookup will indicate which level of attribute to select for each attribute decision, or the first Prob_{jk}, as shown in Eq. (8).

For the first decision, random number 0.282 is drawn. This value falls within the interval $[13.04, 64.52]$ for L2, so the "color blue" is selected. Our second decision random number is 0.964. This value falls within the interval $[62.48, 100]$ for L3, so the "large size" is selected. The first ant, selected L2 (blue) for attribute 1 and L3 (large) for attribute 2. When a path is taken and decisions are made, the history matrix is updated: Updated $\mathbf{G}$ matrix $= \begin{bmatrix} 1 & \mathbf{2} & 1 \\ 1 & 1 & \mathbf{2} \end{bmatrix}$, showing paths chosen incremented by 1. The history matrix now provides information to update the global probabilities, which is analogous to an ant laying a pheromone trail for future ants. GlobProb_{jk} is a pure derivation of this matrix; the values are shown in Table 12. This update will enhance the paths just taken.

Local updates, on the other hand, are designed to lessen the probabilities for paths not taken. A user specified parameter, η, determines the degree of local updating. For our simple problem, an $\eta = 0.95$ has

Table 11
Sample problem cumulative probabilities

	Color (Attribute 1)			Size (Attribute 2)		
	Red (L1)	Blue (L2)	Green (L3)	Small (L1)	Medium (L2)	Large (L3)
Probability (%)	13.03	51.49	35.48	37.53	24.95	37.53
Cume Prob. (%)	13.03	64.52	100.00	37.53	62.47	100.00
Prob. interval	[0, 13.03]	[13.04, 64.52]	[64.53, 100]	[0, 37.53]	[37.54, 62.47]	[62.48, 100]

Table 12
Sample problem global probabilities

Attribute	Level 1	Level 2	Level 3
1	25%	50%	25%
2	25%	25%	50%

been selected. With each subsequent iteration, a path not selected will be decreased by a probability of five percentage points. Table 13 shows the process of local probability updates, from original matrix to updated matrix, to recalculated matrix. In the original matrix, L2 for Attribute 1 and L3 for Attribute 2 are safe from local updates because these are the chosen paths. Each of the other path probabilities will be reduced by 5%, as shown in the updated matrix values.

It should be noted that the sum probabilities for the updated matrix do not sum to 100%. Therefore, these probabilities must be re-weighted, giving the recalculated matrix of Table 14. In this example, local updates do two things: it minimizes probabilities on paths not recently taken, and in the process, reinforces due to re-weighting, the relative probabilities of paths that are taken.

There is now sufficient information to calculate the decision probabilities for the second ant, as indicated in Table 14. The probabilities are calculated using Eq. (8), taking all three types of probabilities into consideration. The data for this table are extracted from Tables 6–8. For example, Attribute 1, Level 1 is shown as 50.72 (highlighted); or the sum of 13.03 (NPV), 25.00 (global) and 12.69 (local). Because we are adding the sum of three sets of probabilities, the sum total for this matrix is 300. Prob_{jk} values are calculated by rescaling these values on a 100-point scale. These probabilities represent the decision space for the next ant, or the next iteration. The probability of the next ant choosing level 1 of attribute 1, Prob_{11}, is 16.91%; Prob_{12} is 15.42% and Prob_{13} is 31.67%. When an ant makes a series of decisions and formulates a solution, a Z value is reported.

Table 13
Sample problem local probability updates

	L1	L2	L3	Sum Prob
Original matrix				
Attribute 1 (%)	13.03	**51.49**	35.48	100.00
Attribute 2 (%)	37.53	24.95	**37.53**	100.00
Updated matrix				
Attribute 1	12.38	**51.49**	33.70	97.57
Attribute 2	35.65	23.70	**37.53**	96.88
Recalculated matrix				
Attribute 1 (%)	12.69	**52.77**	34.54	100.00
Attribute 2 (%)	36.80	24.47	**38.74**	100.00

Table 14
Sample problem probability matrix

Attribute	L1	L2	L3	Sum
Probability matrix				
1	**50.72**	154.26	95.02	300
2	99.32	74.42	126.26	300
Recalculated probability matrix				
1	16.91%	51.42%	31.67%	100%
2	33.11%	24.81%	42.09%	100%

With each subsequent run, this individual Z value is compared to the global Z^* value. If the last solution is better than the global value, this new Z value becomes the new global Z^*, with global updating reinforcing the path just taken. If the Z value is smaller than the global Z^*, the path will be marked as sub-optimal. The search space is always open to new paths because the initial NPVProb_{jk} values remain unchanged. This helps to minimize the danger of getting stuck in a local optimal path.

This entire process will repeat through Ants number of iterations, when the Z^* should finally stabilize with ''optimal'' paths having extremely high global probabilities.

References

Alexouda, G., Paparrizos, K., 2001. A genetic algorithm approach to the product line design problem using the seller's return criterion: An extensive comparative computational study. European Journal of Operational Research 134, 165–178.

Balakrishnan, S., Jacob, V., 1996. Genetic algorithms for product design. Management Science 42, 1105–1117.

Bonabeau, E., Meyer, C., 2001. Swarm intelligence: A whole new way to think about business. Harvard Business Review 79, 106–114.

Bonabeau, E., Theraulaz, G., 2000. Swarm smarts. Scientific American 282, 72–79.

Bonabeau, E., Dorigo, M., Theraulaz, G., 1999. Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, New York, NY.

Bonabeau, E., Dorigo, M., Theraulaz, G., 2002. Inspiration for optimization from social insect behavior. Nature 406, 39–42.

Brunswik, E., 1955. Representative design and probabilistic theory in functional psychology. Psychological Review 62, 193–217.

Business Week, 1993. Flops: Too many new products fail. August 16, pp. 76–82.

Colorni, A., Dorigo, M., Maniezzo, V., Trubia, M., 1994. Ant system for job-shop scheduling. Belgian Journal of Operations Research, Statistics and Computer Science 34, 39–53.

DiCaro, G., Dorigo, M., 1998. AntNet: Distributed stigmergetic control for communications networks. Journal of Artificial Intelligence Research 9, 317–365.

Dobson, G., Kalish, S., 1988. Positioning and pricing a product line. Marketing Science 7, 107–125.

Dobson, G., Kalish, S., 1993. Heuristics for pricing and positioning a product line using conjoint and cost data. Management Science 39, 160–175.

Dorigo, M., Maniezzo, V., Colorni, A., 1996. The Ant System: Optimization by a Colony of Cooperating Agents. IEEE Transactions on Systems, Man and Cybernetics 26, 29–41.

Dorigo, M., DiCaro, G., Gambardella, L., 1999. Ant colonies for discrete optimization. Artificial Life 5, 137–172.

Fishken, D., 1988. Marketing and cost factors in product optimization. Food Technology (November), 138–140.

Gambardella, L., Dorigo, M., 2000. An ant colony system hybridized with a new local search for the sequential ordering problem. INFORMS Journal on Computing 12, 237–255.

Green, P., Krieger, A., 1985. Models and heuristics for product line selection. Marketing Science 4, 1–19.

Green, P., Krieger, A., 1987. A consumer-based approach to designing product line extensions. Journal of Production Innovation Management 4, 21–32.

Green, P., Krieger, A., 1989. Recent contributions to optimal product positioning and buyer segmentation. European Journal of Operational Research 41, 127–141.

Green, P., Srinivasan, V., 1978. Conjoint analysis in consumer research: Issues and outlook. Journal of Consumer Research 5, 102–123.

Green, P., Srinivasan, V., 1990. Conjoint analysis in marketing: New developments with implications for research and practice. Journal of Marketing 54, 3–19.

Green, P., Wind, Y., 1975. New way to measure consumers' judgments. Harvard Business Review 53, 107–117.
Green, P., Carroll, J., Goldberg, S., 1981. A general approach to product design via conjoint analysis. Journal of Marketing 45, 17–37.
Griffin, A., Hauser, J., 1993. The voice of the consumer. Marketing Science 12, 1–27.
Hauser, J., Simmie, P., 1981. Profit maximizing perceptual positions: An integrated theory for the selection of product features and price. Management Science 27, 33–55.
Hooker, J., 1995. Testing heuristics: We have it all wrong. Journal of Heuristics 1, 33–42.
Kaul, A., Rao, V., 1995. Research for product positioning and design decisions: An integrative review. International Journal of Research in Marketing 12, 293–320.
Kohli, R., Krishnamurti, R., 1987. A heuristic approach to product design. Management Science 33, 1523–1533.
Kohli, R., Krishnamurti, R., 1989. Optimal product design using conjoint analysis: Computational complexity and algorithms. European Journal of Operational Research 40, 186–195.
Kohli, R., Sukumar, R., 1990. Heuristics for product-line design using conjoint analysis. Management Science 36, 1464–1478.
Lehmann, D., 1971. Evaluating marketing strategy in a multiple brand market. Journal of Business Administration 3, 15–26.
McBride, R., Zufryden, F., 1988. An integer programming approach to the optimal product line selection problem. Marketing Science 7, 126–140.
McMullen, P.M., Tarasewich, P., Nair, S., 2001. Using swarm intelligence to solve the product line design problem. In: Proceedings of the Decision Sciences Institute. Decision Sciences Institute, San Francisco, CA.
Moore, W., Winer, R., 1987. A panel-data based method for merging joint space and market response function estimation. Marketing Science 6, 25–47.
Nair, S., Thakur, L., Wen, K.-W., 1995. Near optimal solutions for product line design and selection: Beam search heuristics. Management Science 41, 767–785.
Neter, J., Kutner, M., Nachtsheim, C., Wasserman, W., 1996. Applied Linear Regression Models, fourth ed. McGraw-Hill, Boston, MA.
Quelch, J., Kenny, D., 1994. Extend profits, not product lines. Harvard Business Review 72, 154–160.
Shocker, A., Srinivasan, V., 1974. A consumer-based methodology for the identification of new product ideas. Management Science 20, 921–937.
Shocker, A., Srinivasan, V., 1979. Multiattribute approaches for product concept evaluations and generation: A critical review. Journal of Marketing Research 16, 159–180.
Urban, G., Hauser, J., 1993. Design and Marketing of New Products, second ed. Prentice Hall, Englewood Cliffs, NJ.
Zufryden, F., 1977. A conjoint measurement-based approach for optimal new product design and market segmentation. In: Shocker, A.D. (Ed.), Analytic Approaches to Product and Market Planning. Marketing Science Institute, Cambridge, MA, pp. 100–114.