

Review of Marketing Science Working Papers

Volume 1, Issue 4

2002

Working Paper 4

A Probabilistic One-Step Approach to the Optimal Product Line Design Problem Using Conjoint and Cost Data

Winfried J. Steiner
University of Regensburg

Harald Hruschka
University of Regensburg

Review of Marketing Science Working Papers is produced by The Berkeley Electronic Press
(bepress). <http://www.bepress.com/roms>

Copyright ©2002 by the authors.

The author retains all rights to this working paper.

A Probabilistic One-Step Approach to the Optimal Product Line Design Problem Using Conjoint and Cost Data

Dr. Winfried J. Steiner

Graduate School of Economics, Department of Marketing, University of Regensburg,
Universitätsstraße 31, 93053 Regensburg, Germany
winfried.steiner@wiwi.uni-regensburg.de

Prof. Dr. Harald Hruschka

Graduate School of Economics, Department of Marketing, University of Regensburg,
Universitätsstraße 31, 93053 Regensburg, Germany

Department Head
harald.hruschka@wiwi.uni-regensburg.de

March 2002

Correspondence to:

Winfried J. Steiner

Phone (0941) 943-2274

Fax (0941) 943-2277

E-mail:winfried.steiner@wiwi.uni-regensburg.de

A Probabilistic One-Step Approach to the Optimal Product Line Design

Problem

Using Conjoint and Cost Data

Long Abstract: Designing and pricing new products is one of the most critical activities for a firm, and it is well-known that taking into account consumer preferences for design decisions is essential for products later to be successful in a competitive environment (e.g., Urban and Hauser 1993). Consequently, measuring consumer preferences among multiattribute alternatives has been a primary concern in marketing research as well, and among many methodologies developed, conjoint analysis (Green and Rao 1971) has turned out to be one of the most widely used preference-based techniques for identifying and evaluating new product concepts. Moreover, a number of conjoint-based models with special focus on mathematical programming techniques for optimal product (line) design have been proposed (e.g., Zufryden 1977, 1982, Green and Krieger 1985, 1987b, 1992, Kohli and Krishnamurti 1987, Kohli and Sukumar 1990, Dobson and Kalish 1988, 1993, Balakrishnan and Jacob 1996, Chen and Hausman 2000). These models are directed at determining *optimal* product concepts using consumers' idiosyncratic or segment level part-worth preference functions estimated previously within a conjoint framework.

Recently, Balakrishnan and Jacob (1996) have proposed the use of Genetic Algorithms (GA) to solve the problem of identifying a share maximizing *single product* design using conjoint data. In this paper, we follow Balakrishnan and Jacob's idea and employ and evaluate the GA approach with regard to the problem of optimal *product line* design. Similar to the approaches of Kohli and Sukumar (1990) and Nair et al. (1995), product lines are constructed directly from part-worths

data obtained by conjoint analysis, which can be characterized as a one-step approach to product line design. In contrast, a two-step approach would start by first reducing the total set of feasible product profiles to a smaller set of promising items (reference set of candidate items) from which the products that constitute a product line are selected in a second step. Two-step approaches or partial models for either the first or second stage in this context have been proposed by Green and Krieger (1985, 1987a, 1987b, 1989), McBride and Zufryden (1988), Dobson and Kalish (1988, 1993) and, more recently, by Chen and Hausman (2000).

Heretofore, with the only exception of Chen and Hausman's (2000) probabilistic model, all contributors to the literature on conjoint-based product line design have employed a deterministic, first-choice model of idiosyncratic preferences. Accordingly, a consumer is assumed to choose from her/his choice set the product with maximum perceived utility with certainty. However, the first choice rule seems to be an assumption too rigid for many product categories and individual choice situations, as the analyst often won't be in a position to control for all relevant variables influencing consumer behavior (e.g., situational factors). Therefore, in agreement with Chen and Hausman (2000), we incorporate a probabilistic choice rule to provide a more flexible representation of the consumer decision making process and start from segment-specific conjoint models of the conditional multinomial logit type. Favoring the multinomial logit model doesn't imply rejection of the widespread max-utility rule, as the MNL includes the option of mimicking this first choice rule.

We further consider profit as a firm's economic criterion to evaluate decisions and introduce fixed and variable costs for each product profile. However, the proposed methodology is flexible enough to accommodate for other goals like market share (as well as for any other probabilistic

choice rule). This model flexibility is provided by the implemented Genetic Algorithm as the underlying solver for the resulting nonlinear integer programming problem. Genetic Algorithms merely use objective function information (in the present context on expected profits of feasible product line solutions) and are easily adjustable to different objectives without the need for major algorithmic modifications.

To assess the performance of the GA methodology for the product line design problem, we employ sensitivity analysis and Monte Carlo simulation. Sensitivity analysis is carried out to study the performance of the Genetic Algorithm w.r.t. varying GA parameter values (population size, crossover probability, mutation rate) and to finetune these values in order to provide near optimal solutions. Based on more than 1500 sensitivity runs applied to different problem sizes ranging from 12.650 to 10.586.800 feasible product line candidate solutions, we can recommend: (a) as expected, that a larger problem size be accompanied by a larger population size, with a minimum popsize of 130 for small problems and a minimum popsize of 250 for large problems, (b) a crossover probability of at least 0.9 and (c) an unexpectedly high mutation rate of 0.05 for small/medium-sized problems and a mutation rate in the order of 0.01 for large problem sizes.

Following the results of the sensitivity analysis, we evaluated the GA performance for a large set of systematically varying market scenarios and associated problem sizes. We generated problems using a 4-factorial experimental design which varied by the number of attributes, number of levels in each attribute, number of items to be introduced by a new seller and number of competing firms except the new seller. The results of the Monte Carlo study with a total of 276 data sets that were analyzed show that the GA works efficiently in both providing near optimal product line solutions and CPU time. Particularly, (a) the worst-case performance ratio of the GA

observed in a single run was 96.66%, indicating that the profit of the best product line solution found by the GA was never less than 96.66% of the profit of the optimal product line, (b) the hit ratio of identifying the optimal solution was 84.78% (234 out of 276 cases) and (c) it tooks at most 30 seconds for the GA to converge. Considering the option of Genetic Algorithms for repeated runs with (slightly) changed parameter settings and/or different initial populations (as opposed to many other heuristics) further improves the chances of finding the optimal solution.

Key words: Conjoint Analysis, Product Line Design, Probabilistic Choice Modeling, Genetic Algorithms

A Probabilistic One-Step Approach to the Optimal Product Line Design

Problem

Using Conjoint and Cost Data

1 Introduction

Designing and pricing new products is one of the most critical activities for a firm, and it is well-known that taking into account consumer preferences for design decisions is essential for products later to be successful in a competitive environment (e.g., Urban and Hauser 1993). Consequently, measuring consumer preferences among multiattribute alternatives has been a primary concern in marketing research as well, and among many methods developed, conjoint analysis (Green and Rao 1971) has turned out to be one of the the most widely used preference-based techniques for identifying and evaluating new product concepts. This is reflected by a huge number of contributions in the marketing literature that have been devoted to both theoretical advances (for reviews, see Green and Srinivasan 1978, 1990, Green and Krieger 1996) and practical applications (e.g., see Cattin and Wittink 1982, Wittink and Cattin 1989) of conjoint analysis. Moreover, a number of conjoint-based models with special focus on mathematical programming techniques for optimal product (line) design have been developed (e.g., Zufryden 1977, 1982, Green and Krieger 1985, 1987b, 1992, Kohli and Krishnamurti 1987, Kohli and Sukumar 1990, Dobson and Kalish 1988, 1993, Balakrishnan and Jacob 1996, Chen and Hausman 2000). These models seek to determine *optimal* product concepts using consumers' idiosyncratic or segment level part-worth preference functions estimated previously within a conjoint framework.

Optimal product design models proposed so far can be classified according to the following three criteria: (1) the underlying objective function (maximizing profit, share of choices or welfare), (2) the type of choice rule employed (deterministic or probabilistic) and (3) whether only one or multiple items are considered for introduction or modification (single product or product line). As conjoint-based searching for optimal product designs results in combinatorial optimization problems because of the typically discrete nature of attributes used in conjoint studies, and nearly all of these problems are known to be mathematically intractable or NP-hard, mainly heuristic solution procedures have been proposed for the various problem types (for a comprehensive review of research in marketing on optimal product (line) design, see Kaul and Rao 1995).

In the following, we focus on the *product line* design problem. As compared to almost all previous contributions on optimal product line design in which the first choice rule is used to model consumers' choices, we start from segment-specific conjoint models of the conditional multinomial logit type (CMNL) and therefore incorporate probabilistic choice. To the best of our knowledge, only Chen and Hausman (2000) to date have employed a probabilistic choice rule in the context of conjoint-based optimal product line design but their approach (as opposed to ours) requires a reference set of candidate products from which the new product line is selected to be predetermined. We further consider profit contribution as a seller's economic criterion to evaluate decisions. However, the proposed methodology is flexible enough to accommodate for other goals like market share. This model flexibility is provided by the use of a Genetic Algorithm (GA) that is employed to solve the resulting nonlinear integer programming problem. Genetic Algorithms, a probabilistic search technique from the field of artificial intelligence research, merely use objective function information (in the present context about expected profits of feasible product line solutions), and are easily adjustable to different objectives without the

need for major algorithmic modifications. An important feature of the GA approach is that it allows for constructing product line candidates directly from attribute level part-worths data which is preferable to reference set enumeration if the number of attributes and attribute levels is large and most multi-attribute items represented by different attribute level combinations are economically and technologically feasible (Kohli and Sukumar 1990, Nair et al. 1995). With our application of the GA methodology to product line design, we follow Balakrishnan and Jacob (1996) who recently introduced the use of Genetic Algorithms to the product design literature. They have dealt with the problem of identifying a share maximizing new single product design and have shown their algorithm to be of excellent performance (with an average 99,13% close-to-optimal ratio across 192 data sets).

The paper is organized as follows; in §2, we first briefly review some basic aspects of optimal product (line) design concerning the conjoint measurement methodology and choice modeling issues. It follows an overview of hitherto proposed profit-oriented models for product line design and selection. In §3, the new conjoint-based probabilistic product line design model is formalized as a nonlinear integer programming problem with special focus on profit maximizing firms. After a short introduction into the basic GA process, we then present the genetic algorithm developed for solving the seller's problem. We further discuss some problems of GA implementation arising specifically with *product line* design. In §4, we describe the experimental methodology used to evaluate the performance of the proposed GA methodology and present the associated results. §5 summarizes the contents of the paper and draws an outlook onto future research perspectives.

2 Background

2.1 Customer Preference Measurement

Preference-based product (line) design requires customers' preferences to be determined in the run-up. In the present context of conjoint analysis, one first selects attributes considered relevant in the customers' eyes, and a discrete number of feasible levels is fixed for each attribute. The next step is to collect scaled preference evaluations from respondents with regard to a subset of multi-attribute product profiles (stimuli) constructed according to a fractional factorial design. From these preference data, idiosyncratic part-worth preference functions are estimated for each respondent applying decompositional methods (typically OLS regression). Alternatively, attribute level part-worths can be computed from respondents' simulated choice data which is then called a choice-based conjoint analysis (CBC) and establishes a direct connection between preference and choice (e.g., see Louviere and Woodworth 1983). CBC involves the specification of a discrete choice model (like CMNL) and is usually conducted at the aggregate level resulting in pooled attribute level parameter estimations. No matter how, as the part-worth utilities have been estimated, composite utilities for any feasible product profile constructable from the underlying attribute levels can be predicted and used to evaluating new product concepts. Typically, only main effects (and, sometimes, a few two-way interaction effects) are estimated in conjoint studies to limit the loss of predictive power of the model from estimating too many parameters (Green and Srinivasan 1990, Green and Krieger 1996).

As Wittink and Cattin (1989) have reported from a survey on commercial applications of conjoint analysis, market segmentation ranks among the primary purposes of suppliers in conjoint studies. If segmentation issues are of particular interest, individual level part-worth estimations might further be clustered to form market segments (post hoc segmentation).

Moreover, a number of procedures for simultaneously performing market segmentation and calibrating segment-level part-worths in conjoint analysis have been developed in recent years. Such methods for simultaneous segmentation and estimation have been proposed for both the traditional conjoint and the CBC approach (see Wedel and Kamakura 1998 for a comprehensive review).

2.2 Choice Modeling

To model consumers' choices, one needs to specify both a preference model (e.g., a main-effects part-worth model) and a choice rule. While the first one defines the functional relationship between attribute values of a product and a consumer's or a segment's overall utility attached to it, the latter relates preference to choice.

Under a deterministic, first choice rule of preferences, a consumer is assumed to choose from her/his choice set the product with the highest associated utility with certainty. Consequently, an individual is expected to switch to a new product if it offers to her/him a higher utility than her/his current favorite brand. However, the first choice rule seems to represent an assumption too restrictive for many product categories and individual choice situations, as the analyst is possibly not able to consider all variables that influence consumer behavior (e.g., situational factors) and then cannot infer actual choice from preference with certainty. As a result, applying the first choice rule improperly leads to suboptimal results on the aggregate market level, as market shares of products with higher utilities across consumers would be overestimated.

Consequently, the use of a probabilistic choice rule can often provide a more realistic representation of the consumer decision making process (e.g., see Kaul and Rao 1995). Moreover, some probabilistic choice rules (like the ones discussed below) offer high flexibility in

calibrating actual choice behavior including the option of mimicking the first choice rule. Reviewing the literature on optimal product design, two probabilistic choice rules have been employed so far: the generalized (or powered) Bradley-Terry-Luce share-of-utility rule (GBTL, α -rule) and the logit choice rule (CMNL).

According to the GBTL model, a consumer i 's (segment i 's) choice probability P_{ij} w.r.t. a product j ($j=1,\dots,J$) is defined by the ratio of its associated deterministic utility V_{ij} to the sum of associated deterministic utilities for the various alternatives considered for buying:

$$P_{ij} = \frac{V_{ij}^{\alpha}}{\sum_{m=1}^J V_{im}^{\alpha}} \quad (1)$$

The GBTL model can be calibrated on actual market shares by post hoc optimization of the decision constant alpha (e.g., see Green and Krieger 1993). With $\alpha \rightarrow \infty$, GBTL approximates the first-choice rule, and with $\alpha = 1$, the model mimics the traditional BTL share-of-utility rule.

Starting from the assumption of independently and identically extreme value type I distributed error terms, one arrives at the logit choice rule, a discrete choice model (McFadden 1974). Considering a conditional multinomial logit (CMNL), the probability P_{ij} that a consumer i (a segment i 's consumer) will choose brand j from a set of J alternatives is:

$$P_{ij} = \frac{\exp(\mu \cdot V_{ij})}{\sum_{m=1}^J \exp(\mu \cdot V_{im})} \quad (2)$$

Like with GBTL, calibration on actual market shares can be carried out subsequently to preference estimation by post hoc optimization of the scaling parameter μ (e.g., Choi and

DeSarbo 1993). As μ goes to infinity, the logit behaves like a deterministic model, and as μ approaches zero, it becomes a uniform distribution. However, discrete choice models are best suited to estimate consumers' preferences directly from choice data (e.g., see Green and Krieger 1996). In this case, preference estimation and model calibration perform simultaneously and tests for statistical inferences about a particular model and its parameters are available. Then, the scaling parameter μ is absorbed by the other parameters of V_{ij} .

2.3 Profit-Oriented Approaches to Product Line Design

A number of researchers have proposed (part-worth) utility-based procedures for selecting a product line maximizing a seller's profit. These approaches can be classified into two categories. *One-step approaches* solve the problem by constructing product lines directly from part-worth preference and cost/return functions. *Two-step approaches*, on the other hand, start by reducing the total set of feasible product profiles to a smaller set of promising items (reference set of candidate items) from which the products that constitute a product line are selected in a second step with the objective of maximum profit contribution. The final product line decision in the second step is made on the basis of total utility and total profit of each reference set item (as opposed to part-worth preferences and costs/returns at the individual attribute level).

Most researchers dealing with the two-step approach have introduced partial models which are limited to the second step, i.e., the determination of a product line from a reference set of candidate items (e.g., Green and Krieger 1985, McBride and Zufryden 1988, Dobson and Kalish 1988, 1993, Chen and Hausman 2000). Mainly, greedy and greedy-interchange heuristics have been proposed to solve these second step problems. Only Green and Krieger (1987a, 1987b, 1989) have also considered the question of how to generate such a reference set in an appropriate

way and have presented several heuristic procedures. Kohli and Sukumar (1990) and Nair et al. (1995), on the other hand, have proposed one-step approaches to optimal product line design. Kohli and Sukumar solve the seller's return problem via a dynamic programming heuristic, using attributes as stages and attribute levels as states, whereas Nair et al. have employed a beam search solution technique which originates from artificial intelligence.

Two-step approaches are known to work well with problems in which the reference set contains a small number of candidate items or most product profiles in larger problems are technologically and economically infeasible. Otherwise, a one-step approach is preferable, as the intermediate step of enumerating utilities and profits of a huge number of reference set items could then be eliminated. However, as mentioned in the beginning, a substantial deficiency of presented approaches to optimal product line design and selection (except the recently developed model of Chen and Hausman 2000) is the assumption of a deterministic, first choice model of consumer choice. Whereas Chen and Hausman (2000) have closed this gap w.r.t. the two-step approach by presenting a probabilistic model for the second step problem, we now present a probabilistic one-step approach to the optimal product line design problem maximizing a seller's profit. Afterwards, we propose the use and assess the performance of Genetic Algorithms (GA) to solve the problem.

3 Description of Problem

As typical for conjoint-based product (line) design models, the utility function is assumed to be an additive main effects part-worth model. We further model consumer behavior at the segment level which has become very popular in recent years (see section 2.1), although one could also assume individual level part-worth utilities without loss of generality. Thus, from a seller's point

of view who wants to launch a new product line consisting of R new items, the conjoint utility function can be specified as follows:

$$V_{ir} = \sum_{k \in \hat{K}} \sum_{l \in \hat{L}_k} \lambda_{ikl} e_{klr}, \quad (i \in \hat{I}, r \in \hat{R}) \quad (3)$$

where

\hat{I} : set of segments ($i = 1, \dots, I$);

\hat{K} : set of relevant attributes ($k = 1, \dots, K + 1$) with price as attribute ($K + 1$);

\hat{L}_k : set of feasible levels of attribute k ($l = 1, \dots, L_k$);

\hat{R} : set of new items to be selected by the seller ($r = 1, \dots, R$);

V_{ir} : segment i 's (deterministic) utility for the seller's item r ;

λ_{ikl} : segment i 's part-worth utility with respect to level l of attribute k ;

e_{klr} : a (0,1) variable that equals 1 if level l of attribute k is assigned to the seller's item r .

Using the logit choice rule (CMNL) and taking into account existing brands of competitors, the choice probability of (an individual of) segment i w.r.t. the seller's item $r' \in \hat{R}$ is:

$$P_{ir'} = \frac{\exp(\mu \cdot V_{ir'})}{\sum_{r \in \hat{R}} \exp(\mu \cdot V_{ir}) + \sum_{j \in \hat{J}} \exp(\mu \cdot V_{ij})} \quad (i \in \hat{I}, r' \in \hat{R}) \quad (4)$$

where

\hat{J} : set of existing competitive brands ($j = 1, \dots, J$);

$P_{ir'}$: probability that (a consumer of) segment i will choose the seller's item r' ;

μ : scaling parameter of the CMNL model ($\mu > 0$).

Assuming profit maximization as the seller's goal, variable and fixed costs are to be included into the model as well. Following the SIMOPT model of Green and Krieger (1992), the variable unit cost function is assumed to be a linear-additive model that can be formulated as follows:

$$c_r^{(\text{var})}(\bar{e}_{\text{klr}}) = \sum_{k=1}^K \sum_{l=1}^{L_k} c_{\text{kl}}^{(\text{var})} e_{\text{klr}}, \quad (r \in \hat{R}) \quad (5)$$

where

$c_{\text{kl}}^{(\text{var})}$: the seller's variable cost for level l of attribute k ;

\bar{e}_{klr} : a (0,1) design vector of length $L = \sum_{k=1}^K L_k$, indicating the presence/absence of levels of the non-price attributes with respect to the seller's item r ;

$c_r^{(\text{var})}(\bar{e}_{\text{klr}})$: variable unit cost for the seller's item r , represented by profile \bar{e}_{klr} .

Consequently, variable costs are assumed to be available (estimable) at the individual attribute level which is quite a realistic premise if the seller has an operating cost accounting system.

Likewise, a linear-additive fixed cost function $c_r^{(\text{fix})}(\bar{e}_{\text{klr}}) = \sum_{k=1}^K \sum_{l=1}^{L_k} c_{\text{kl}}^{(\text{fix})} e_{\text{klr}}$ is employed providing the opportunity to assign fixed costs to item r if these do also depend on its profile \bar{e}_{klr} .

Notes on allocating fixed costs to products can be found in Dobson and Kalish (1993) and Moore et al. (1999).

Let further $\bar{e}_{(K+1)r}$ be a (0,1) vector of length L_{K+1} indicating the presence or absence of price level l w.r.t. the seller's item r , and let $\bar{p} = [p_1, \dots, p_{L_{K+1}}]$ be the vector of feasible price levels, then $p_r = \bar{p} \otimes \bar{e}_{(K+1)r}$ is the price assigned to item r . Finally, let Q_i denote the size of segment i , then the seller's problem of designing a profit maximizing product line becomes the following nonlinear programming problem:

Maximize

$$\sum_{r \in \hat{R}} \left[(p_r - c_r^{(\text{var})}(\vec{e}_{\text{klr}})) \cdot \sum_{i \in \hat{I}} Q_i \cdot P_{ir} - c_r^{(\text{fix})} \right] \quad (6)$$

subject to

$$\sum_{l \in \hat{L}_k} e_{\text{klr}} = 1, \quad (k \in \hat{K}, r \in \hat{R}), \quad (7)$$

$$\sum_{k \in \hat{K}} \sum_{l \in \hat{L}_k} |e_{\text{klr}} - e_{\text{klr}'}| > 0, \quad (r, r' \in \hat{R}, r \neq r'), \quad (8)$$

$$e_{\text{klr}} \in \{0, 1\}, \quad (k \in \hat{K}, l \in \hat{L}_k, r \in \hat{R}). \quad (9)$$

Objective function (6) maximizes the total profit the seller obtains by offering a product line of R items. Constraint (7) requires that exactly one level of each attribute is assigned to each item (exclusiveness condition). Constraint (8) ensures that several items of the seller pairwise must differ in at least one attribute level (divergence condition). Finally, constraint (9) represents the binary restrictions with regard to the decision variables of the optimization problem.

4 Application of Genetic Algorithms to Product Line Design

Genetic Algorithms (GA), first proposed by Holland (1975), are based on the principle of natural selection which results in ‘survival of the fittest’. Recently, Balakrishnan and Jacob (1996) introduced the use of GA to the product design literature. They have dealt with the problem of finding a share maximizing new *single* product assuming a deterministic first choice behavior. As their GA implementation has shown to be of excellent performance (with an average 99.13% close-to-optimal ratio across 192 data sets), we now propose and evaluate the GA approach with respect to the above stated problem of optimal *product line* design.

GAs work on a coding of strings (chromosomes), particular string positions or substrings corresponding to variables (genes) which in turn can take on a number of values (alleles). Each string as a whole then represents a candidate solution of the underlying optimization problem. To connect the GA approach with our model, we consider a candidate product line solution to be specified in a binary string format. With R items to be selected by the seller ($r = 1, \dots, R$), $K+1$ attributes in the product category ($k = 1, \dots, K+1$) and L_k levels of attribute k ($l = 1, \dots, L_k$), a string is defined to be composed of $R \cdot (K+1)$ substrings where substring $k + r \cdot (K+1)$ corresponds to attribute k of item r and is made up of L_k binary string positions reflecting the feasible levels of attribute k . Consequently, a one in a specific substring denotes the presence of a specific attribute level implying the other $L_k - 1$ substring positions (i.e., attribute levels) to be zero at the same time.

Following the basic GA process (e.g., see Goldberg 1989, Michalewicz 1996), the seller's problem (6)-(9) is solved iteratively in the following way. First, an initial population P_0 of G strings is randomly generated and each string is assigned a fitness value which corresponds to its profit value obtained by calculating (6), respectively. Then, evolution (optimization) starts by applying the standard genetic operators reproduction, crossover and mutation to create successively new generations of offspring (P_t , $t = 1, 2, \dots$). By reproduction, strings are copied according to their fitness, i.e., strings with higher profit values are granted a higher probability to participate in the creation of offspring reflecting the idea of the 'survival of the fittest'. To operationalize the reproduction operator, we employ a binary tournament selection procedure (Dawid 1996): $G/2$ pairs of strings are randomly chosen from the actual population (with

replacement) and from each pair of strings only the product line candidate solution with higher profit evaluation is selected for mating.

After reproduction, each two of the parent strings in the mating pool are picked randomly (without replacement) and each pair of strings undergoes crossover with probability p_{cross} . Specifically, crossover proceeds at the substring level (because the unit of interest is the attribute) in exchanging a number of substrings between two parent strings leading to two new product line candidate solutions. We use simple one-point crossover by randomly fixing one cross site and then swapping the partial strings to the right of this crossover point. With R new items and $K+1$ attributes, there are $R \cdot (K+1) - 1$ feasible cross sites.

Subsequent to crossover, each offspring is provided a chance to mutate. Like crossover, mutation acts at the substring level by picking a single attribute with probability p_{mut} and then by altering the corresponding attribute level within this substring at random. Mutation is known as a background operator, as mutation rates too high would disturb the search process and would lead to some kind of random search. Empirical findings indicate a mutation probability p_{mut} on the order of one mutation per thousand “bits” as a rule of thumb to obtain good solutions (e.g., Goldberg 1989, Dawid 1996). Whereas reproduction reduces the diversity in the population, mutation maintains a certain degree of heterogeneity of string solutions which is necessary to avoid premature convergence of the GA process.

Once the transition process from P_t to P_{t+1} is completed, the newly generated product line candidates are assigned their associated fitness values. Each time a new generation is created, the algorithm further checks whether or not an underlying stopping condition is met. Following Balakrishnan and Jacob (1996), we use a moving average rule as stopping criterion, as they have

shown this rule to provide a good indication of convergence to a solution. Specifically, the GA process terminates if the average fitness of the three best strings of the current generation has increased by less than $x\%$ (convergence rate) as compared to the average fitness of the three best strings over the three previous generations. We set the convergence rate to 0.2% to impose a sufficiently strong condition to convergence (Balakrishnan and Jacob 1996). Finally, in each generation the highest fitness value achieved so far and its corresponding string are updated and stored to make sure that the best product line solution found over all generations (and not only of the final generation) is returned at convergence. The GA implementation is fully described in Appendix A.

We now discuss how the divergence condition (8) is implemented by the GA. Randomly generating an initial population can be repeated until the divergence condition is fulfilled by each string. The same principle can be used for crossover by repeatedly searching for one cross site until admissible offsprings are obtained. However, interchanging segments of two admissible parents may theoretically lead to at least one offspring violating the divergence condition. A way to deal with this problem is to accept the respective parents as members of the next generation without modifying them. To this end, we add an exit condition if a maximum number of crossover repetitions is exceeded. Mutation does not cause similar problems, as the random recoding of a substring does not necessarily result in a different binary substring vector. That is, for each of $R \cdot (K + 1)$ substrings of an admissible offspring resulting from crossover, at least the given binary substring vector fulfills the divergence condition, respectively. Nevertheless, to avoid that an admissible binary coding is not found for the respective substring after repeated mutation, we have employed a maximum number of repeated mutations as another exit condition.

After convergence, the GA returns the product line with the highest fitness (profit) as well as related profits and segment-specific market shares of each of the new items. Moreover, intermediate results of each generation (like product line candidates and their fitness values) and some descriptive statistics (like number of crossovers and mutations, average population fitness, population standard deviation and best product line solution found so far) can be viewed. This way the decision maker may analyze the development of the GA and may have a look at other feasible product line solutions with high fitness evaluations.

5 Performance Evaluation

In this section, we evaluate the performance of the proposed GA by means of sensitivity analysis and Monte Carlo simulation. First, we study the sensitivity of the approximation of optimal solutions w.r.t. varying parameter values (population size, crossover probability, mutation rate) for different problem sizes. Second, based on the results of the sensitivity analysis, we employ a Monte Carlo simulation to obtain evidence on the approximation behavior and CPU time requirements of the GA. In particular, we provide degrees of approximation (relative to optimal solutions), hit rates and CPU times across a large set of problem instances.

5.1 Parameter Selection for the Genetic Algorithm

To recommend parameter values for population size, crossover and mutation probabilities which provide near optimal solutions we analyze sensitivity of GA solutions for various product market specifications with different problem sizes. This analysis may show that different problem sizes require different parameter values (e.g., it can be expected that population size varies with the number of feasible product line solutions, hence, with the size of the search space).

Hypothetical data used for GA configuration refer to the product category ‘sneakers’ and are based on sellers’ catalogues and a survey of retailing selling personnel. Attributes considered are price, cushioning system, stability and upper with 5, 5, 4 and 4 attribute levels, respectively. Part-worth utilities of the corresponding attribute levels are generated in a way to reflect differences between up to four segments (A, B, C, D), e.g., with respect to price sensitivity (for more details, see Appendix B).

To simplify interpretation of results, we assume for both sensitivity analysis and Monte Carlo simulation that (a) segment sizes Q_i are identical and (2) fixed costs do not vary across feasible product profiles. Based on these assumptions optimal solutions are determined for problems with 2 attributes (price and cushioning system), 3 attributes (price, cushioning system and stability) and 4 attributes (price, cushioning system, stability and upper) which are specified in table 1. The number of feasible product line solutions (i.e., the size of the search space) depends on the number $K+1$ of product attributes, the number L_k of levels of attribute k and the number R of new items to be introduced by the seller in the following manner:

$$\binom{\prod_{k=1}^{K+1} L_k}{R} \quad (10)$$

Table 1: Number of Product Line Candidate Solutions

New Items	Attributes	Problem Size	Segment Combination
2	4***	79.800	BC*****
3	3**	161.700	ABC*****
3	4***	10.586.800	BCD*****
4	3**	3.921.225	ABCD
4	2*	12.650	ABCD

-
- * Price and Cushioning System (5 Levels Each)
 - ** Price, Cushioning System and Stability (2×5 Levels, 1×4 Levels)
 - *** Price, Cushioning System, Stability and Upper (2×5 Levels, 2×4 Levels)
 - **** Chosen at Random Among the 4 Segments (A, B, C, D)
-

The four segment combinations (BC, ABC, BCD, ABCD) represent the product markets analyzed in the various optimization runs. Problems are defined in such a way that their optimal solution can be found by complete enumeration. This allows to examine how well the solutions determined heuristically by the GA approximate the optimal solutions.

Sensitivity analysis is based on a $12 \times 5 \times 3$ factorial design with 12 values of population size (G) in the range [30; 250], at increments of 20 strings, 5 values of crossover probability p_{cross} (0.6; 0.7; 0.8; 0.9; 1.0) and 3 different mutation rates p_{mut} (0.0; 0.01; 0.05). Values of crossover and mutation probabilities are set in accordance with experiences gained in previous GA applications which suggest a crossover probability of at least 0.6 and a very low mutation rate.¹ Because of the lack of comparable results, more values of population size are studied. Computations are performed for two randomly generated starting configurations of existing items of incumbent firms for each of the five problems contained in table 1.

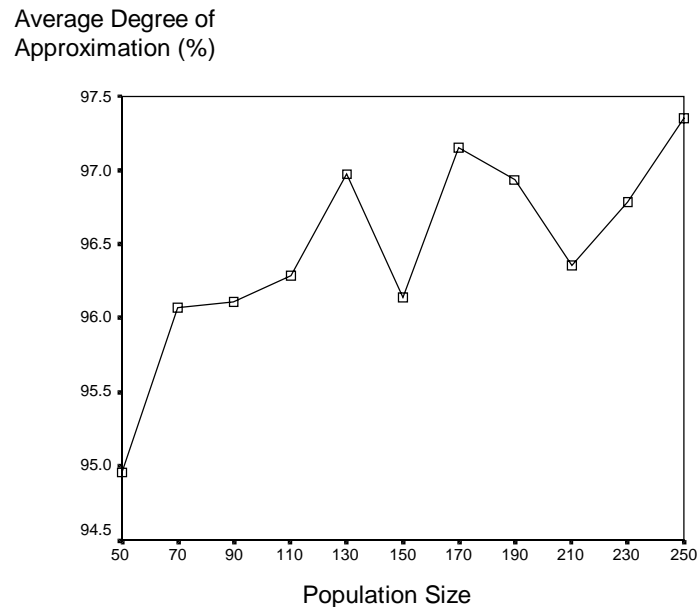
Table 2 shows recommended parameter values for the GA on the basis of more than 1500 test runs and associated average degrees of approximation (Avg_Appr) of the optimal solutions (e.g., for problem 4 the highest average degree of approximation w.r.t. crossover was achieved for a crossover probability of 1.0).² For the smaller problems 1 and 2, even population sizes of 130 and 150, respectively, lead to very high degrees of approximation. That is why we do without test runs with higher populations sizes for these problems.

Table 2: Recommended GA Parameter Values

No.	Problem Size	G	Avg_App r	p _{cross}	Avg_App r	p _{mut}	Avg_App r	Test Runs
1	12.650	130	99.5%	1.0	99.0%	0.05	98.9%	180
2	79.800	150	99.0%	0.9	98.4%	0.05	98.8%	300
3	161.700	230	98.3%	1.0	97.6%	0.05	97.7%	360
4	3.921.225	250	99.2%	1.0	98.6%	0.01	98.5%	360
5	10.586.800	250	97.5%	1.0	96.8%	0.01	96.8%	360

As expected, a larger problem size (search space) requires a larger population size in order to capture the higher diversity of product line candidate solutions. In other words, a sufficient degree of heterogeneity of string solutions (especially w.r.t. the initial population) is necessary to guarantee that the solution space is explored thoroughly and a satisfactory high degree of approximation can be attained. On the other hand, a population size which is too large increases CPU time, but may improve approximation to only a modest extent. Similar to the elbow criterion known from cluster analysis, population sizes can be set to a value after which improvements of approximation level off. Especially for problems of type 5, even higher values of population size ($G > 250$) can be expected to lead to still better approximations (see figure 1).

Figure 1: Average Approximation Levels for Problem Type 5 Depending on Population Size



Unequivocal recommendations can be given for crossover probabilities. Without exception, crossover probabilities greater equal 0.9 (in four out of five cases even equal to 1.0) are appropriate. On the whole, we see a clear tendency that higher crossover probabilities lead to better approximations.

Results w.r.t. mutation rates are somewhat surprising. For the small and medium problems 1, 2 and 3 the recommended value of 0.05 is unexpectedly high (despite the fact that mutation acts at the substring level). For the larger problems 4 and 5, on the other hand, high mutation rates have a negative effect on approximation and seem to disturb the search process by putting too much weight on the random component. For large problems the recommended mutation rate of 1% at the substring level lies within the usual range.

5.2 Monte Carlo Simulation

In view of our experiences with the GA w.r.t. sensitivity to changes in parameter values, we employed a Monte Carlo study to assess the GA performance for a large set of systematically

varying market scenarios and associated problem sizes. We generated problems using a $3 \times 3 \times 3 \times 3$ factorial experimental design which varied by the number of attributes (including price) in the product category (3, 4, 5), number of levels in each attribute (2, 3, 4), number of items to be introduced by a new seller (2, 3, 4) and number of competitors (1, 2, 3) except the new seller. In order not to go beyond the scope of the study, we further coupled the number of segments as well as the number of existing items of the incumbent firms to the number of new items to be introduced by the seller.³ Of the 81 possible problems, we solved a subset of 69 problems. The remaining problems are not solved because of exorbitant CPU time requirements for complete enumeration which is once again used to identify optimal product line solutions. Four replications were solved for each of the 69 problems leading to a total of 276 data sets that were analyzed.

For each replication, segment-specific part-worth utilities and variable costs of attribute levels were generated randomly from uniform distributions. Moreover, starting configurations of existing items of competitors were fixed at random each time. As for most products consumers are price-sensitive, we additionally ensured that higher price levels were assigned lower part-worth utilities.

Following the results of the GA configuration, we set GA parameter values w.r.t. to the various problems depending on the respective problem size, as is shown in table 3:

Table 3: GA Parameter Settings for Different Problem Sizes (Compare Table 2)

Problem size	G	p_{cross}	p_{mut}
≤ 12.650	130	1.0	0.05
$]12.650; 79.800]$	150	0.9	0.05

]79.800; 161.700]	230	1.0	0.05
]161.700; 3.921.225]	250	1.0	0.01

Table 4 gives an overview of the Monte Carlo study and shows the associated simulation results. For a clear representation, we summarized results from 12 simulation runs w.r.t. each problem size (4 replications with 1, 2 or 3 competitors, respectively). Remember that the problem size which results according to expression (10) does not depend on the number of competing firms in a product market.

As the simulation results indicate, the GA provided near optimal solutions for nearly all data sets analyzed, with a worst-case average performance ratio of 99.55% w.r.t. the various problem sizes

Table 4: Experimental Design and Simulation Results (Monte Carlo Study)

Market Parameters			Parameter Values of the GA				Generations (= Iterations)			Ratio of GA Solution Value to Optimal Solution Value (%) (Performance Ratio)		Average CPU Time		
# Attr	# Lev	# Items	Problem Size	Pop	Co	Mut	Min/Ma x	Avg	σ (Gen)	MinRatio (%)	AvgRatio (%)	σ (Appr)	Hits	Sec
3	2	2	28	130	1.0	0.05	3/4	3.17	0.39	100	100	0.00	12	1
3	2	3	56	130	1.0	0.05	3/5	3.33	0.65	100	100	0.00	12	2
3	2	4	70	130	1.0	0.05	3/6	4.25	0.97	100	100	0.00	12	4
3	3	2	351	130	1.0	0.05	5/8	6.33	0.89	100	100	0.00	12	1
3	3	3	2.925	130	1.0	0.05	6/9	8	1.04	100	100	0.00	12	4
3	3	4	17.550	150	1.0	0.05	4/13	8.25	1.87	98.39	99.83	0.46	9	7
3	4	2	2.016	130	1.0	0.05	6/11	7.92	1.50	100	100	0.00	12	2
3	4	3	41.664	150	1.0	0.05	7/17	10.25	2.96	100	100	0.00	12	5
3	4	4	635.376	250	1.0	0.01	7/14	10.83	1.95	98.41	99.74	0.45	6	16
4	2	2	120	130	1.0	0.05	3/7	4.25	1.36	100	100	0.00	12	1
4	2	3	560	130	1.0	0.05	3/8	5.83	1.53	100	100	0.00	12	3
4	2	4	1.820	130	1.0	0.05	4/10	6.50	2.19	99.57	99.96	0.12	11	7
4	3	2	3.240	130	1.0	0.05	5/13	8.66	2.61	100	100	0.00	12	2
4	3	3	85.320	230	1.0	0.05	4/18	10.66	3.82	97.49	99.55	0.78	7	9
4	3	4	1.663.740	250	1.0	0.01	7/17	13.16	2.41	98.79	99.70	0.38	4	24
4	4	2	32.640	150	1.0	0.05	6/15	10.58	2.64	98.46	99.84	0.45	10	3
4	4	3	2.736.520	250	1.0	0.01	8/14	11.75	1.76	98.72	99.76	0.43	8	11
5	2	2	496	130	1.0	0.05	4/9	5.91	1.44	100	100	0.00	12	2
5	2	3	4.960	130	1.0	0.05	3/12	7.83	2.33	98.32	99.86	0.48	11	5
5	2	4	35.960	150	1.0	0.05	6/15	9.66	2.71	98.55	99.87	0.42	10	12
5	3	2	29.403	150	1.0	0.05	7/19	13.25	4.14	96.66	99.55	1.00	9	4
5	3	3	2.362.041	250	1.0	0.01	6/17	12.08	3.82	97.49	99.65	0.74	8	12
5	4	2	523.776	250	1.0	0.05	10/21	14.83	3.88	97.99	99.67	0.73	9	9

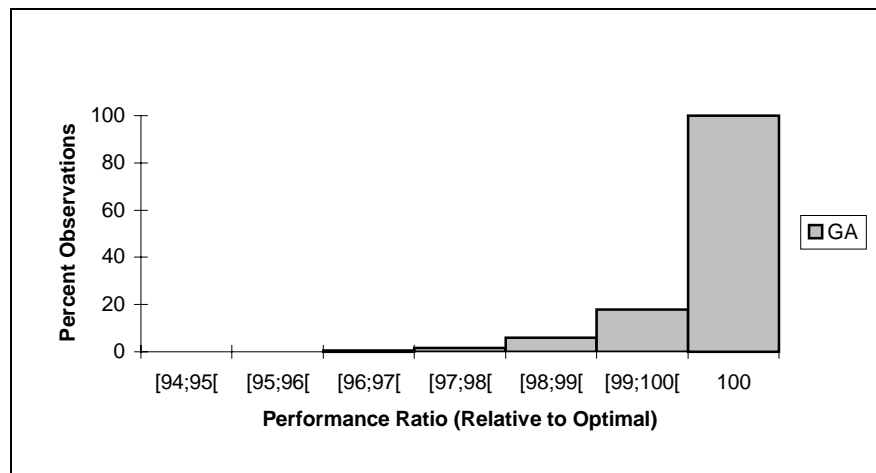
GA: Genetic Algorithm; **# Attr/Lev/Items:** Number of Attributes/Attribute Levels/Items to Introduce; **Pop/Co/Mut:** Population Size/Crossover Probability/Mutation Rate;

Min/Max/Avg: Minimum/Maximum/Average Number of Generations (= Iterations) until Convergence; **σ (Gen):** Standard Deviation of Generations (= Iterations) until Convergence;

MinRatio (%)/**AvgRatio (%)**: Minimum/Average Ratio of the Best Product Line Identified by the GA to the Profit-Maximizing Product Line (Average Performance Ratio refers to 12 Replications);
 σ (Appr): Standard Deviation of GA Performance Ratio (for 12 Replications); **Hits**: Number of Cases (of 12 Replications) in which the optimal solution was identified by the GA;

(i.e., averaged across 12 replications, respectively) and a minimum performance ratio of 96.66 % in a single run. That is, the profit of the best product line found by the GA across 276 data sets was never less than 96.66 % of the profit of the optimal product line. In addition, the standard deviations σ (Appr) of performance ratios w.r.t. the various problems always lie within a 1% range indicating a high robustness of the GA. Figure 2 shows the fraction of problems with a performance ratio within a specified interval and the number of cases in which the optimal solution was found by the GA. The performance ratio was at least 99% in 272 cases (98% in 244 cases) out of a total of 276 cases and the hit rate was 84.78% (i.e., the optimal solution was identified in 234 cases). Depending on problem size, it tooks on average between 3.17 and 14.83 generations and not more than 30 seconds in a single run for the GA to reach convergence.⁴

Figure 2: Histogram of Performance Ratio Across 276 Simulated Problems



Analyses of variances were performed to assess the impact of the four factors (number of attributes, number of attribute levels, number of new items, number of competitors) on both the number of generations and CPU time. W.r.t. the number of generations, only main effects turned out to be statistically significant (with an overall R^2 of 0.64) indicating that the GA requires

more generations to converge as the number of attributes, the number of attribute levels or the number of new items to be introduced increases ($p < 0.001$, respectively). However, the number of new items is clearly of less importance (with an associated eta-squared value of $\eta = 0.124$) than the number of attributes and attribute levels ($\eta = 0.418$; $\eta = 0.571$), and the number of competitors had no significant main effect ($p > 0.05$). W.r.t. CPU times, all four main effects ($p < 0.001$, respectively) and even all two-way interactions ($p < 0.025$, respectively) proved to be statistically significant resulting in an R^2 of 0.872. Among the main effects, the number of new items shows the strongest influence on CPU times ($\eta = 0.796$), whereas the number of competitors and related interaction terms are by far of least importance.

6 Conclusions

In this paper, we proposed a new probabilistic approach to the optimal product line design problem using conjoint and cost data. The model allows the inclusion of consumers' preferences, counterpart products of competitors as well as variable and fixed costs. Product lines are constructed directly from attribute level part-worths utilities and attribute level costs. To model consumers' choices, we employed segment-specific conjoint models of the conditional multinomial logit type, but it would also be possible to start from part-worths data estimated at the individual level and/or to incorporate another probabilistic choice rule like the generalized Bradley-Terry-Luce share-of-utility rule (GBTL). We developed and applied a genetic algorithm to solve the optimal product line design problem and carried out sensitivity analysis and Monte Carlo simulation to assess the performance of the GA methodology. Similar to results obtained by Balakrishnan and Jacob (1996) who solved the problem of identifying a share maximizing single product design via Genetic Algorithms, our study indicates that the GA works efficiently

in both providing near optimal product line solutions (with a worst-case solution of 96.66% relative to the optimal) and CPU time (with a maximum CPU time of 30 seconds in a single run). Although we used our model to solve the seller's problem of introducing a new product line with the objective of maximum profit contribution, the proposed framework could easily be adjusted to handle the less complex problem of maximizing share-of-choices and/or the problem of extending an existing product line (i.e., allowing for the case of already existing items owned by a seller). This is supported by the high flexibility of the GA which merely uses objective function information and, therefore, could accommodate for different fitness criteria without the need for (major) algorithmic modifications.

If interactions between attributes are to be considered, the additive main-effects utility model, as defined by expression (1), could easily be extended to that effect. Including interaction terms would not require the specification of additional decision variables. Interaction terms would merely have an effect on a string's fitness evaluation by taking into account the associated part-worth utilities for preference evaluation. A way to deal with technological infeasibility of attribute level combinations would be to incorporate related interaction terms into the variable or fixed cost functions and to penalize them with high cost values.

An important feature of Genetic Algorithms is their ability to carry out repeated runs with (slightly) changed parameter values and/or different initial populations, thus improving the chances of finding the optimal or at least a near optimal solution. As Balakrishnan and Jacob (1996) have already pointed out, another important characteristic of the GA approach is that solutions obtained from other techniques can be inserted in the initial population. Thus, rather than generating all the members of the initial population at random, the GA could use knowledge

about potential optima in arranging the initial population and improve on an existing solution which then defines a kind of lower bound or benchmark for GA performance. The GA methodology also provides high flexibility with regard to the final product line decision, as the decision maker may be provided by quite a number of solutions with similar high fitness values just as she/he wants. This way, the decision maker can additionally impose, e.g., strategic fit criteria for selecting the best product line.

For future research, this approach needs to be extended to consider retaliatory responses from incumbent firms. This may be verified by explicitly modeling competitive reactions within a game theoretic framework. For single brand firms, Choi and DeSarbo (1993) and Green and Krieger (1997) have already worked in this direction and have illustrated how to derive competitive strategies in conjoint analysis under the Nash equilibrium concept.

Footnotes

- ¹ Mutation rates analyzed for sensitivity refer to the substring level and, consequently, are higher than in most GA applications in which mutation applies to individual string positions.
- ² The corresponding average value of 98.6% is the arithmetic mean of approximations attained over 36 runs of the GA (12 values of population size times three values of mutation rates).
- ³ For example, if two new items are to be introduced, we assume a product market to consist of two segments and two items currently being offered by each of the incumbent firms.
- ⁴ The current implementation is on a 366 MHz personal computer.

Appendix A: Description of the Genetic Algorithm

1. Initialization

Set $\tau := 0$ and randomly generate an initial population P_0 consisting of G strings with length $L = R \cdot \sum_{k=1}^{K+1} L_k$ which all satisfy constraints (7), (8) and (9). Determine fitness values (profits) of these strings (product line candidate solutions) according to objective function (6).

2. Generations (Iterations)

Repeat until the average fitness of the three best strings of generation τ has increased by less than $x\%$ (convergence rate) as compared to the moving average fitness (i.e., the average fitness of the three best strings of generations $\tau - 1$, $\tau - 2$, $\tau - 3$):

(a) Reproduction (Binary Selection)

Randomly select (with replacement) two strings out of the G members of generation P_τ and choose from this pair of strings the one with higher fitness to become a member of the mating pool. Repeat this selection process G times (= population size).

(b) Crossover (One-Point Crossover)

Randomly pick (without replacement) each two of the reproduced parent strings of the mating pool and cross each pair with probability p_{cross} . Crossover proceeds at the substring level by randomly fixing one of $R \cdot (K + 1)$ feasible cross sites and swapping the partial strings to the right of the crossover point leading to two new offsprings.

Repeat crossover for any selected pair until both offsprings fulfill the divergence condition (8) or a maximum number of repeated crossovers is exceeded. In the latter case, accept both parent strings unmodified as members of generation $\tau + 1$.

(c) Mutation

Replace each of $R \cdot (K + 1)$ substrings of each of the G offsprings with probability p_{mut} by a new, randomly generated binary vector which fulfills the exclusivness condition (7).

Repeat mutation until divergence condition (8) is fulfilled or a maximum number of repeated mutations is exceeded.

(d) Fitness Evaluation

Set $\tau := \tau + 1$ and determine the fitness values of the newly generated product line candidate solutions of generation $P_{\tau+1}$ according to objective function (6). Store the best string found so far together with its associated fitness.

Appendix B: Part-Worth Utilities and Variable Cost Data Used for GA Configuration

Table 5 shows product attributes and feasible attribute levels. *Figure 3* illustrates part-worth utilities of attribute levels for each of four segments. *Table 6* contains variable cost data at the individual attribute level for the non-price attributes.

Table 5: Product Attributes and Attribute Levels in the Product Category Sneakers

Attributes	Levels	Shortcut
1. Price	\$45	45
	\$55	55
	\$65	65
	\$75	75
	\$85	85
2. Cushioning System	Leightweight Cushioning	LEI
	Air Cushioning	AIR
	Hexalite Cushioning	HEX
	Gel Cushioning	GEL
	Variable Fit Cushioning	VAR
3. Stability	a. Base Support *	BAS
	b. Rearfoot/Forefoot Stability	RFS
	c. Pronation/Supination Control	PSC
	d. Full Support	FUL
4. Upper	a. Nylon	NYL
	b. Synthetic Leather	SYN
	c. Mesh	MES
	d. Leather	LEA

* without exceptional stability and motion control features

Figure 3: Representation of Segment Level Part-Worth Utilities λ_{ikl}

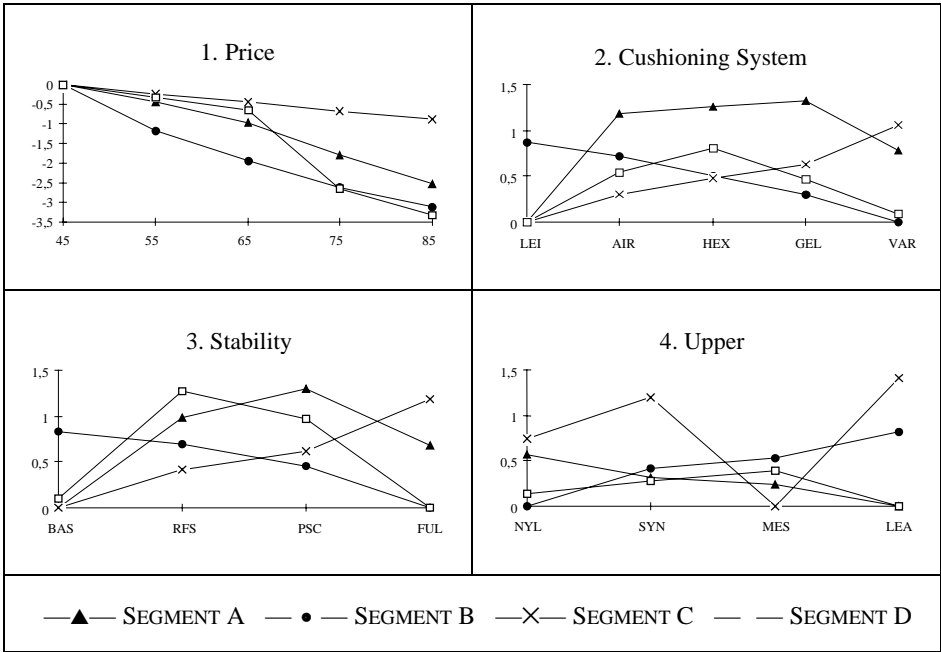


Table 6: Variable Costs (\$) at the Individual Attribute Level Depending on the Number of Non-Price Attributes Considered (1, 2 or 3)

Attributes	Cushioning System					Stability				Upper			
Levels	LEI	AIR	HEX	GEL	VAR	BAS	RFS	PSC	FUL	NYL	SYN	MES	LEA
Number of Attributes in the Product Category (except Price)													
1 Attribute	26	43	44.5	46	57								
2 Attributes	12.5	21.5	22.5	23.5	29	13	21	24.5	28.5				
3 Attributes	9.5	16	16.5	17	21.5	10	15.5	18	21	7	11	13.5	15

References

- Balakrishnan, P.V. and Jacob V.S. (1996). Genetic algorithms for product design. *Management Science*, Vol. 42 (1), 1996. pp. 1105–1117.
- Cattin, P. and Wittink, D.R. (1982). Commercial use of conjoint analysis: A survey. *Journal of Marketing*, Vol. 46(3), 1982. pp. 44–53.
- Chen, K.D. and Hausman W.H. (2000). Technical note: Mathematical properties of the optimal product line selection problem using choice-based conjoint analysis. *Management Science*, Vol. 46(2), 2000. pp. 327–332.
- Choi, S.C. and DeSarbo, W.S. (1993). Game theoretic derivations of competitive strategies in conjoint analysis. *Marketing Letters*, Vol. 4, 1993, pp. 337–348.
- Dawid, H. (1996). *Adaptive learning by genetic algorithms: Analytical results and applications to economic models*, Springer, Berlin, 1996.
- Dobson, G. and Kalish, S. (1988). Positioning and pricing a product line. *Marketing Science*, Vol. 7(2), 1988. pp. 107–125.
- Dobson, G. and Kalish S. (1993). Heuristics for pricing and positioning a product-line using conjoint and cost data. *Management Science*, Vol. 39(2), 1993. pp. 160–175.
- Goldberg, D.E. (1989). *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley, Reading, MA, 1989.
- Green, P.E. and Krieger, A.M. (1985). Models and heuristics for product line selection. *Marketing Science*, Vol. 4, 1985. pp. 1–19.
- Green, P.E. and Krieger, A.M. (1987a). A consumer-based approach to designing product line extensions. *Journal of Product Innovation Management*, Vol. 4, 1987. pp. 21–32.

- Green, P.E. and Krieger, A.M. (1987b). A simple heuristic for selecting ‘good’ products in conjoint analysis. *Applications of Management Science*, Vol. 5, 1987. pp. 131–153.
- Green, P.E. and Krieger, A.M. (1989). Recent contributions to optimal product positioning and buyer segmentation. *European Journal of Operational Research*, Vol. 41, 1989. pp. 127–141.
- Green, P.E. and Krieger, A.M. (1992). An application of a product positioning model to pharmaceutical products. *Marketing Science*, Vol. 11, 1992. pp. 117–132.
- Green, P.E. and Krieger, A.M. (1993). Conjoint analysis with product-positioning applications. In: Eliashberg, J. and Lilien, G.L. (eds.). *Handbooks in OR & MS (5): Marketing*, Elsevier Science, Amsterdam, 1993. pp. 467–515.
- Green, P.E. and Krieger, A.M. (1996). Individualized hybrid models for conjoint analysis. *Management Science*, Vol. 42(6), 1996. pp. 850–867.
- Green, P.E. and Krieger, A.M. (1997). Using conjoint analysis to view competitive interaction through the customer’s eyes. In: Day, G.S. and Reibstein, D.J. (eds.). *Wharton on dynamic competitive strategy*, Wiley & Sons, New York, 1997. pp. 343–368.
- Green, P.E. and Rao, V.R. (1971). Conjoint measurement for quantifying judgmental data. *Journal of Marketing Research*, Vol. 8, 1971. pp. 355–363.
- Green, P. E. and Srinivasan, V. (1978). Conjoint analysis in consumer research: Issues and outlook. *Journal of Consumer Research*, Vol. 5, 1978. pp. 103–123.
- Green, P. E. and Srinivasan, V. (1990). Conjoint analysis in marketing: New developments with implications for research and practice. *Journal of Marketing*, Vol. 54, 1990. pp. 3–19.
- Holland, J.H. (1975). *Adaptation in natural and artificial systems*, The University of Michigan Press, Ann Arbor, MI, 1975.

- Kaul, A. and Rao, V.R. (1995). Research for product positioning and design decisions: An integrative review. *International Journal of Research in Marketing*, Vol. 12, 1995. pp. 293–320.
- Kohli, R. and Krishnamurti, R. (1987). A heuristic approach to product design. *Management Science*, Vol. 33, 1987. pp. 1523–1533.
- Kohli, R. and Sukumar, R. (1990). Heuristics for product-line design using conjoint analysis. *Management Science*, Vol. 36, 1990. pp.1464–1478.
- Louviere, J.J. and Woodworth, G. (1983). Design and analysis of simulated consumer choice or allocation experiments: An approach based on aggregate data. *Journal of Marketing Research*, Vol. 20, 1983. pp. 350–367.
- McBride, R.D. and Zufryden, F.S. (1988). An integer programming approach to the optimal product line selection problem. *Marketing Science*, Vol. 7, 1988. pp.126–140.
- McFadden, D. (1974). Conditional logit analysis of qualitative choice behavior. In: Zarembka, P. (ed.). *Frontiers of Econometrics*, Academic Press, New York, 1974. pp.105–142.
- Michalewicz, Z. (1996). *Genetic algorithms + Data structures = Evolution programs*, Springer, Berlin, 1996.
- Moore, W.L., Louviere, J.J. and Verma, R. (1999). Using conjoint analysis to help design product platforms. *Journal of Product Innovation Management*, Vol. 16, 1999. pp. 27–39.
- Nair, S.K., Thakur, L.S. and Wen, K. (1995). Near optimal solutions for product line design and selection: Beam search heuristics. *Management Science*, Vol. 41(5), 1995. pp. 767–785.
- Urban, G.L. and Hauser, J.R. (1993). *Design and marketing of new products*, Prentice-Hall, New Jersey, 1993.

- Wedel, M. and Kamakura, W.A. (1998). *Market segmentation: Conceptual and methodological foundations*, Kluwer, Massachusetts, 1998.
- Wittink, D.R. and Cattin, P. (1989). Commercial use of conjoint analysis: An update. *Journal of Marketing*, Vol. 53. pp. 91–96.
- Zufryden, F.S. (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, 1977. pp. 100–114.
- Zufryden, F.S. (1982). Product-line optimization by integer programming, Annual ORSA/TIMS Meeting, San Diego, CA, 1982.