European Journal of Operational Research 40 (1989) 186–195
North-Holland

Theory and Methodology

Optimal product design using conjoint analysis: Computational complexity and algorithms *

Rajeev KOHLI
Joseph M. Katz Graduate School of Business, University of Pittsburgh, Pittsburgh, PA, USA

Ramesh KRISHNAMURTI
Simon Fraser University, Burnaby, B.C., Canada

Abstract: The problem of maximizing the share of a new product introduced in a competitive market is shown to be NP-hard. A directed graph representation of the problem is used to construct shortest-path and dynamic-programming heuristics. Both heuristics are shown to have arbitrarily-bad worst-case bounds. Computational experience with real-sized problems is reported. Both heuristics identify near-optimal solutions for the simulated problems, the dynamic-programming heuristic performing better than the shortest-path heuristic.

Keywords: Computational complexity, heuristics, worst case, empirical performance, marketing, conjoint analysis

1. Introduction

The problem of selecting an optimal new product using consumer-preference data has received significant attention in the marketing literature (Green, 1984; Green and Srinivasan, 1978). Generally, consumer preferences are modelled by a linear function in which the attributes are used as the independent variables. For example, if color (green, blue) and size (small, large) are used as attributes in a model of consumer preferences for cars, the preference function may be

$$z = a + bx + cy + e, \tag{1}$$

where z is a preference measure; x and y are 0–1 dummy variables representing the attribute levels (e.g., $x = 0$ (1) if the color is red (green), and $y = 0$ (1) if the size is small (large)); a, b and c are model parameters; and e is an error term. The parameters b and c are called the 'part worths' of attribute levels, and (1) is called a part-worths model (Green and Srinivasan, 1978). Interactions among attributes may also be represented in the linear model, although they generally are not used in practice because the main-effects model performs well in predicting consumer preferences. Ordinal or cardinal preferences over a set of hypothetical multiattribute items are scaled to estimate idiosyncratic preference functions (Kruskal, 1965; Srinivasan and Shocker, 1973). The product profiles evaluated by respondents are usually selected from a fractional factorial experimental plan (e.g., Addelman, 1962; Plackett and Burman, 1946) in

* The authors thank R. Sukumar, University of Pittsburgh, for computer programming assistance. Thanks are also expressed to Monique Speilberg and Paul E. Green at the University of Pennsylvania for their encouragement and helpful comments.

Received September 1987; revised May 1988

0377-2217/89/$3.50

which at the attributes (attribute levels) serve as the design factors (factor levels).

Preference simulations, such as those performed by Green, Carroll and Goldberg's (1981) POSSE models, use individual preference functions to predict the potential performance of a new product concept. A popular 'first choice' model assumes that given a set of alternatives, an individual chooses an item with the highest utility. In a certain 'share-of-choices' formulation, an 'optimal' new product is described as one that maximize the number ('share') of consumers who prefer the selected product concept to (idiosyncratic) status-quo items (Cattin and Wittink, 1982; Green, Carroll and Goldberg, 1981; Green and Srinivasan, 1978). The present paper examines the computational properties and solution procedures for this problem.

Consider a share-of-choices problem with K attributes at n levels each. Selecting an optimal product profile requires evaluating n^K product profiles, a number which increases exponentially with K. The computational time for explicit enumeration increases correspondingly. Consequently, a method more efficient than enumeration is needed for solving share-of-choices problems of practical sizes. Zufryden (1977) describes a mathematical programming formulation of the problem, but does not present explicit solution procedures. The computational complexity of the problem is also not known, so that it is unclear whether an exact polynomial-time algorithm is even feasible for the problem. Two related questions follow:

(i) Can an efficient (i.e., polynomial time) algorithm be devised for the share-of-choices problem?

(ii) If not, what heuristics can be used to solve the problem, and how closely do they approximate the optimal solution?

The two questions are the central focus of the paper. In Section 2, we show that the share-of-choices problem is NP-hard. Efficient exact solution procedures are therefore not likely to be found for the problem (Cook, 1971; Garey and Johnson, 1979; Karp, 1975). Section 3 describes a directed-graph representation and the associated integer-programming formulation of the problem. Section 4 describes two heuristics based on the graph structure. We call these the shortest-path and the dynamic-programming heuristics. Section 5 characterizes the worst-case performance of the heuristics and reports empirical tests of their performance with simulated data.

2. Intractability of the share-of-choices problem

Let $\Theta = \{1, 2, \dots, I\}$ denote the set of I sample consumers whose preferences are used to evaluate product concepts. Let $\Omega = \{1, 2, \dots, K\}$ denote the set of K attributes. Let $\Phi_k = \{1, 2, \dots, J_k\}$ denote the set of J_k levels of attribute $k \in \Omega$.

A *product profile p* consists of one level of each attribute and is represented by the vector $p = (j_1, j_2, \dots, j_K)$ of its K attribute levels. Let w_{ijk} denote the part worth of level $j \in \Phi_k$ of attribute $k \in \Omega$. Let j_{ik}^* denote the level of attribute k that appears in consumer i's status-quo item. Then

$$c_{ijk} = w_{ijk} - w_{ij_{ik}^*k}$$

$$\text{for all } i \in \Theta,\ j \in \Phi_k \text{ and } k \in \Omega \tag{2}$$

is consumer i's *relative part worth* for level j of attribute k (i.e., the part worth of level j relative to the part worth of level j_{ik}^* of attribute k). Because the interval-scaled part worths are invariant to affine transformations, we assume below that they are normalized so that each consumer's part-worths utility for each product profile is (strictly) less than 1. A consequence of the normalization is that the relative part-worths utility of every product profile lies between -1 and 1. Therefore according to the first-choice rule, consumer i prefers product profile $p = (j_1, j_2, \dots, j_K)$ to status quo only if

$$c_i(p) = c_{ij_11} + c_{ij_22} + \cdots + c_{ij_KK} \tag{3}$$

is positive. The share-of-choices problem is equivalent to identifying a product profile p for which $c_i(p)$ is positive for a maximum number of consumers $i \in \Theta$.

Theorem 1. *The share-of-choices problem is NP-hard.*

Proof. It is sufficient to show that the following decision problem is NP-complete: Does there exist a product profile p for which the relative part-worths utility $c_i(p)$ is positive for at least R ($\leqslant I$) consumers in Θ?

Let $\Theta_1 = \{i \mid c_i > 0,\ i \in \Theta\}$ be the subset of consumers for whom $c_i(p)$ is positive for product profile p. Then product profile p 'solves' the share-of-choices problem if and only if $I_1 \geqslant R$, where I_1 is the cardinality of Θ_1. We consider $R = I$ and $R < I$ as separate cases. The proof consists of showing that for $R = I$, satisfiability reduces to an instance of the share-of-choices problem, and that for $R <$ I, max-2 satisfiability reduces to an instance of the share-of-choices problem. As satisfiability and max-2 satisfiability are NP-complete (Garey, Johnson and Stockmeyer, 1976; Even, Itai and Shamir, 1976) and reduce to instances of the share-of-choices problem in polynomial time, the latter problem is also NP-complete.

Case 1. $R = I$. Consider the share-of-choices problem in which

(i) each attribute has 2 levels (i.e., $J_k = 2$ for all $k \in \Omega$),

(ii) all part worths are either 0 or 1 (i.e., $w_{ijk} = 0, 1$ for all $i \in \Theta$, $j \in \Phi_k$ and $k \in \Omega$), and

(iii) the part worth of each attribute level in every consumer's status-quo item is zero (i.e., $w^*_{ijk} = 0$ for all $i \in \Theta$, $j \in \Phi_k$ and $k \in \Omega$).

Then $i \in \Omega_1$ if and only if $w_{ijk} > 0$ for at least one level j_k of product profile $(j_i, j_2, \ldots, j_K)$. We show that satisfiability corresponds to this instance of the share-of-choices problem.

Let $u_1, u_2, \ldots, u_K$ be propositional variables. Let $C = \{c_1, c_2, \ldots, c_I\}$ be a set of I clauses described in terms of the propositional variables or their negations $(\bar{u}_k)$. Let a_{k1} denote level 1 of attribute k and let a_{k2} denote level 2 of attribute k. Define the following mapping $f: u_k \to a_k$:

$$\begin{aligned} f(u_k) &= a_{k1}, \\ f(\bar{u}_k) &= a_{k2}, \end{aligned} \quad \text{for all } k \in \Omega. \tag{4}$$

For all $i \in \Theta$ and $k \in \Omega$, define a mapping g from clauses c_i to part worths so that the part worth of level j $(j = 1, 2)$ of attribute k is 1 if clause c_i contains u_k $(\bar{u}_k)$:

$$\begin{aligned} g\colon\ & u_k \in c_i && \leftrightarrow\ w_{i1k} = 1 \text{ and } w_{i2k} = 0, \\ g\colon\ & \bar{u}_k \in c_i && \leftrightarrow\ w_{i1k} = 0 \text{ and } w_{i2k} = 1, \\ g\colon\ & u_k, \bar{u}_k \notin c_i && \leftrightarrow\ w_{i1k} = w_{i2k} = 0, \end{aligned} \tag{5}$$

where $\leftrightarrow$ denotes the mapping induced by g. Then for clause c_i and a product profile $(j_1, j_2, \ldots, j_K)$:

(i) c_i false implies $w_{ijk} = 0$ for all $k \in \Omega$, which in turn implies that $i \notin \Theta_1$,

(ii) c_i true implies $w_{ijk} > 0$ for some $k \in \Omega$, which in turn implies that $i \in \Theta_1$.

Thus c_i is true for all $c_i \in C$ (i.e., C is satisfiable) if and only if $\Theta_1 = \Theta$. The transformation of satisfiability to the share-of-choices problem is complete, the assignment of truth values to u_k corresponding to the selection a level for each attribute.

Case 2. $R < I$. The proof is identical to that for Case 1 except that every clause consists of two literals. The statements (i) and (ii) in the first part of the proof imply that the statement "at least R clauses in C are true" is equivalent to $I_1 \geqslant R$.

Finally, the transformation of both satisfiability and max-2 satisfiability to the share-of-choices problem is achieved in polynomial time because

(i) f is an identity transformation from propositional variables to attribute levels, and

(ii) g is a one-to-one mapping from propositional variables in clauses to part-worths of attribute levels. For S literals per clause, the mapping g requires $2S$ assignments, which is a linear function of S.

3. Graph representation and formulation

Let $1, 2, \ldots, K$ be any arbitrary ordering of the K attributes in Ω. Represent the set of distinct product profiles in a directed graph as follows. From a dummy source node (s), draw forward arcs to J_1 nodes, each associated with a distinct levels of attribute 1. From each of these nodes, draw forward arcs to J_2 nodes, each associated with a distinct level of attribute 2. Continue until all levels of all K attributes are so represented. Terminate the graph by drawing forward arcs into a dummy destination node (d) from each node corresponding to a level of attribute K, the last attribute represented in the graph.

Let $V = \{jk \mid j \in \Phi_k,\ k \in \Omega\}$ denote the set of nodes, where $v = jk$ denotes the node associated with level $j \in \Phi_k$ of attribute $k \in \Omega$. Let $E = \{f_e\}$ denote the set of arcs. With each arc f_e originating from node $v' = j'k'$ and terminating in node $v = jk$, associate the weight $c_{ie} = c_{iv} = c_{ijk}$ (i.e., associate the part worth of the attribute level corresponding to the node in which arc f_e terminates).[1] Then (a) each distinct path from s to d denotes a distinct product profile, and

(b) the sum of arc weights along a path from s to d equals the relative part-worths utility of the associated product profile.

Let A denote the node–arc incidence matrix, its rows corresponding to nodes and its column to arcs. The first two rows of A correspond to the source (s) and destination (d) nodes, respectively. The remaining rows correspond to the nodes associated with attribute levels. The ve-th element of A is

$$a_{ve} = \begin{cases} 1 & \text{if arc } f_e \text{ departs from node } v, \\ -1 & \text{if arc } f_e \text{ terminates in node } v, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

Let f denote a column vector of the arcs ordered like the columns in A. Maximizing share of choices can then be formulated as the following 0–1 integer program:

$$\text{Minimize} \quad z = \sum_{i \in \Theta_2} x_i \quad (7)$$

Subject to

$$\sum_{e \in E} c_{ie} f_e + x_i \geq \delta \quad \text{for all } i \in \Theta_2, \quad (8)$$

$$A \cdot f = [1 \quad -1 \quad 0 \quad \ldots \quad 0]^{\mathrm{T}}, \quad (9)$$

$$x_i = 0, 1 \text{ (integer)} \quad \text{for all } i \in \Theta_2, \quad (10)$$

$$f_e = 0, 1 \text{ (integer)} \quad \text{for all } e \in E, \quad (11)$$

where δ is a small, positive number. Together with the integrality constraint on the f_e, the first (second) row of (9) implies that exactly one arc must depart from s (enter into d). The remaining rows imply that if an arc enters a node, then another arc must also depart from the node. Thus constraint (9) requires that one path (product profile) be selected. The term $\sum_{e \in E} c_{ie} f_e$ in (8) represents the sum of arc weights (relative part worths) along the selected path for consumer i. As $-1 < \sum_{e \in E} c_{ie} f_e < 1$, constraint (8) requires that

$$x_i = \begin{cases} 0 \text{ or } 1 & \text{if } \sum_{e \in E} c_{ie} f_e > 0, \\ 1 & \text{otherwise.} \end{cases} \quad (12)$$

The minimization objective forces $x_i = 0$ whenever $\sum_{e \in E} c_{ie} f_e > 0$, so that x_i counts whether (0) or not (1) consumer i prefers the selected product profile to status quo. As desired, the objective function minimizes (maximizes) the number of consumers for whom the selected product profile is less preferred than (preferred to) status quo.

A procedure for eliminating infeasible solutions is discussed in the next section. Cannibalization of a firm's share (but not profits) from existing brands is controlled by defining θ to be the set of consumers whose status-quo brand is not offered by the firm. Idiosyncratic importance weights reflecting differences in purchase or consumption rates are incorporated in the above formulation by weighting the x_i in the objective function by w_i. In the following discussion, we restrict our discussion to equal consumer weights. The weighting of consumers is incorporated in the following solution procedures in an obvious manner.

4. Heuristics

As the share-of-choices problem cannot be efficiently solved by an exact algorithm, we investigate heuristic solution procedures. A shortest-path heuristic and a dynamic-programming heuristic are considered below. Both are based on the preceding graph representation of product profiles. The shortest-path heuristic simplifies the multi-individual problem to a single-individual problem by aggregating part worths across consumers. It then selects a path for which the sum of the aggregate path worths is maximized. The dynamic-programming heuristic treats attributes as stages and attribute levels as states. Forward recursion is used to construct a path from the source to the destination, the 'best' partial path into each state (attribute level) being identified by augmenting the 'best' paths into each state (level) of the preceding stage (attribute).

[1] The following discussion assumes a main-effects model of preferences. Two-way interactions between pairs of successive attributes in the graph are incorporated by suitably altering the arc weights. Observe that the first (last) attribute may interact only with the second (second-last) attribute in the graph. Intermediate attributes may interact with at most two (preceding and succeeding) attribute in the graph. Although uncommon in practice, more complex two-way and higher order interactions may be represented by defining 'composite' attributes (Green and Srinivasan, 1978).

Infeasible solutions are eliminated by constructing multiple paths, the number of paths being determined as follows. Let N denote the number of infeasible product profiles. Let level j of attribute k appear in N_{jk} ($\leqslant N$) infeasible product profiles. Let N_k be the maximum of the number of infeasible product profiles across the J_k levels of attribute; i.e.,

$$N_k = \max\{ N_{jk} \mid j \in \Phi_k \}. \tag{13}$$

Let $N_{\min} = \min\{ N_k \mid k \in \Omega \}$ be the minimum value of N_k across all attributes $k \in \Omega$, and let k' be a (not necessarily unique) attribute for which $N_{k'} = N_{\min}$. Consider M ($> N_{\min}$) product profiles, all of which have level j of attribute k'. Because each level of attribute k' appears in no more than $N_{\min}$ infeasible product profiles, at least $M - N_{\min}$ of these M product profiles must be feasible. Therefore if M product profiles are identified for *each* level of attribute k', then at least $M - N_{\min}$ product profiles must be feasible for each level of attribute k'.

To ensure a feasible solution, attribute k' (for which $N_{k'} = N_{\min}$) is considered *last* in the following algorithms and M paths terminating in each level of attribute k' are identified. Because each level of attribute k' appears in no more than $N_{\min}$ product profiles, at least $M - N_{\min}$ product profiles associated with each of its levels must be feasible. A single, feasible, product profile is ensured if $M = N_{\min} + 1$. More generally, R feasible product profiles are ensured if $M = N_{\min} + R$.

The shortest-path heuristic

Let $1, 2, \ldots, K$ denote an ordering of the attributes so that $N_K = N_{\min}$. Let $c_{jk} = -\Sigma_{i \in \Theta} c_{ijk}$ denote the sum across consumers of the part worth of level $j \in \Phi_k$ of attribute k. The heuristic selects M paths from s to d along which the sum of the c_{ijk} is the smallest (equivalently, it selects M paths along which the sum of $-c_{jk} = \Sigma_{i \in \Theta} c_{ijk}$ is the largest). Conceptually, the heuristic simplifies a problem involving I individuals to one in which only one dummy individual (with relative part worths c_{jk}) needs to be considered. Although standard procedures for solving the shortest-path problem can be employed (Papadimitriou and Steiglitz, 1982), we use the following dynamic program that exploits the special structure of the graph to solve the problem in $O[JKM(J + \log M)]$, where J denotes the average number of levels across attributes (see Appendix).

Initialization. Set $k = 1$, $S_{j1} = \{ c_{j1} \}$ for all $j \in \Phi_1$.

Recursion. For all $k = 2, \ldots, K$, $j \in \Phi_k$, identify the subset of at most $M = N_{\min} + 1$ largest elements in the set

$$S_{jk} = \{ c_{jk} + s \mid s \in S_{j',k-1},\ j' \in \Phi_{k-1} \}.$$

Termination. Stop if $k = K$. Each subsets S_{jK}, $j \in \Phi_k$, contains M elements, each of which is the sum across individuals of the part-worths utility of a product profile. Because each level of attribute K appears in no more than $N_{\min}$ product profiles, at least one product profile identified for each level of attribute K must be feasible. Let s_{jK} denote the subset of elements in S_{jK} associated with feasible product profiles. The heuristic solution is identified by the largest element in the set $S = \{ s_{jK} \mid j \in \Phi_k \}$.

The dynamic-programming heuristic

Let $C(k)$ denote the individuals-by-attribute levels matrix of part worths for attribute $k \in \Omega$. Each row of $C(k)$ corresponds a distinct consumer, each column to a distinct level of attribute k, and the ij-th element to the relative part worth consumer i associates with level j of attribute k. Let $C_j(k)$ denote the j-th column of $C(k)$. The heuristic mimics a dynamic program in which the attributes are treated as stages and the attribute levels are treated as states. Forward recursion is used to implement the algorithm. We first describe the algorithm for selecting a single product profile, then describe its extension for selecting M product profiles. Conceptually, a path is constructed from the source node (s) into each state of the k-th stage (each level of attribute k, $k = 2, \ldots, K$) by augmenting and selecting one of the n_{k-1} paths that terminate into each state of the $(k-1)$-st stage (each level of attribute $k-1$). The augmented path that is selected maximizes the number of consumers for whom the sum of the c_{ijk} along the path is positive. At the K-th stage, the selected path into each state describes a product profile. The path along which the sum of the c_{ijk} is the positive for the largest number of consumers is selected as the solution. The Appendix shows that the computational complexity of the heuristic is $O[KJM(JI + \log M)]$.

Initialization. Set $k = 1$, $S^*(1) = C(1)$.

Recursion. For all $k = 2$, K, define

$$S_j(k) = S^*(k-1) + [C_j(k)][1] \quad \text{for all } j \in \Phi_k, \tag{14}$$

where [1] is a conformable row vector of unit elements. Let $S_j^*(k)$ denote the column with the largest number of positive elements in $S_j(k)$, and let $S^*(k) = [S_j^*(k)]$. Ties among columns of $S_j(k)$ are broken by selecting the column with the highest number of non-negative elements, and selecting a still-tied column with the largest sum of positive elements. If any columns are still tied, one of them is randomly selected.

Termination. Stop of $k = K$. The heuristic solution is identified by the column of $S^*(K)$ with the largest number of positive elements. The i-th element of $S^*(K)$ denotes consumer i's part-worths utility for the selected product profile.

To eliminate infeasible solutions, the $S^*(k)$ matrix at each step is constructed by selecting M columns, instead of 1 column, from each $S_j(k)$. In all other respects, the preceding algorithm is unchanged.

5. Performance evaluation

Worst-case analysis

Let n(OPT) denote the number of consumers who prefer the optimal product profile to status quo. Let n(SP) [n(DP)] denote the number of consumers who prefer the product profile selected by the shortest-path (dynamic-programming) heuristic to status-quo. To evaluate the worst-case performance of the heuristics, we wish to place lower bounds on the performance ratio

$$r(\text{SP}) = n(\text{SP})/n(\text{OPT}) \tag{15}$$

of the shortest-path heuristic, and

$$r(\text{DP}) = n(\text{DP})/n(\text{OPT}) \tag{16}$$

of the dynamic-programming heuristic.

Table 1
Worst-case for shortest-path heuristic

Consumer	Relative part worths (c_{ijk})			
	Attribute 1		Attribute 2	
	Level 1	Level 2	Level 1	Level 2
1	0	$1-\epsilon$	$-\epsilon$	0
2	ϵ	0	0	$-1+\epsilon$
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
$\frac{1}{2}I+1$	ϵ	0	0	$-1+\epsilon$
$\frac{1}{2}I+2$	0	0	$-\epsilon$	0
$\frac{1}{2}I+3$	0	0	$-\epsilon$	0
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
I	0	0	$-\epsilon$	0

Theorem 2. *The shortest-path heuristic is arbitrarily bad in the worst case.*

Proof. It is sufficient to show the existence of a problem instance for which the performance ratio r(SP) has a worst-case bound of zero. Consider a share-of-choices problem involving two dichotomous attributes and the relative part worths shown in Table 1 for an even number (I) of consumers. It is easy to verify by enumeration that the optimal product profile is described by level 1 of both attributes, and that it is preferred to status quo by consumers 2 through $\frac{1}{2}I + 1$ (i.e., $n(\text{OPT}) = \frac{1}{2}I$). The shortest-path heuristic identifies the product profile described by level 2 of attribute 1 and level 1 of attribute 2. Only consumer 1 prefers this product profile to status quo ($n(\text{SP}) = 1$). Hence

$$r(\text{SP}) = 1/(\tfrac{1}{2}I) = 2/I, \tag{17}$$

and

$$\lim_{I \to \infty} r(\text{SP}) = 0. \tag{18}$$

Theorem 3. *The dynamic-programming heuristic is arbitrarily bad in the worst case.*

Proof. Once again, the proof consists of identifying a problem instance for which the performance ratio r(DP) has a worst-case bound of zero. Consider a share-of-choices problem involving one attributes at three levels, and two dichotomous attributes. Let the relative part worths for I con-

Table 2
Worst-case for dynamic-programming heuristic

Consumer	Relative part worths (c_{ijk})						
	Attribute 1			Attribute 2		Attribute 3	
	Level 1	Level 2	Level 3	Level 1	Level 2	Level 1	Level 2
1	2ϵ	$-\frac{1}{2}-\epsilon$	0	$-\epsilon$	0	$\frac{1}{2}+2\epsilon$	0
2	$-1+\epsilon$	$-\frac{1}{2}-\epsilon$	0	$-\epsilon$	0	$\frac{1}{2}+2\epsilon$	0
3	$-1+\epsilon$	$-\frac{1}{2}-\epsilon$	0	$-\epsilon$	0	$\frac{1}{2}+2\epsilon$	0
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
I	$-1+\epsilon$	$-\frac{1}{2}-\epsilon$	0	$-\epsilon$	0	$\frac{1}{2}+2\epsilon$	0

sumers be shown in Table 2. Then it is easy to verify by enumeration that the optimal product profile is described by levels 2, 2 and 1 of attributes 1, 2 and 3, respectively. All I consumers prefer the optimal product to status quo (i.e., $n(\text{OPT}) = I$). The heuristic identifies the product profile described by the levels 1, 2 and 1 of attributes 1, 2 and 3, respectively. Only consumer 1 prefers this product profile to status quo (i.e., $n(\text{DP}) = 1$). Hence

$$r(\text{DP}) = 1/I, \tag{19}$$

and

$$\lim_{I \to \infty} r(\text{DP}) = 0. \tag{20}$$

Empirical performance of heuristics

The worst-case results suggest that both heuristics can perform poorly for certain problem instances. However, to obtain some insight into their performance with less extreme data, 64 randomly-generated problems were solved. For each problem, the shares of choices for product profiles identified by the shortest-path and the dynamic-programming heuristics were compared to each other and to the shares of choices of the optimal product profile. The problems were generated using a 4^3 experimental design using number of consumers (100, 200, 300, 400), number of attributes (2, 4, 6, 8) and number of levels per attribute (2, 3, 4, 5) as the design factors. Problems involving more than 8 attributes were not solved because the computational time required for their enumeration is very large. For each problem, the part worths were randomly selected from a uniform distribution on [0, 1] and normalized within consumers. A status-quo product profile was randomly specified for each consumer. All product profiles were assumed to be feasible. Computations were performed on a FPS computer. [2]

Table 3 summarizes the simulation results. The dynamic-programming heuristic identifies the optimal solution for 28 (42.5%) problems, the shortest-path heuristic for 9 (14.1%) problems. For 11 problems, the heuristics perform equally well in approximating or identifying the optimal solution. Of the remaining 53 problems, the dynamic-programming heuristic identifies a solution closer to the optimal in 44 (83%) cases.

The cumulative fraction of problems with a performance ratio less than a specified value is

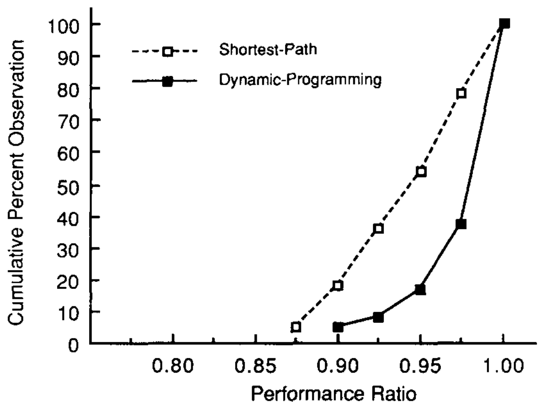

Figure 1. Cumulative distribution of performance ratio across 64 simulated problems for shortest-path and dynamic-programming heuristics

[2] The FPS computer is roughly twice as fast as a VAX-8600 computer and ten times faster than a DEC-10 computer.

Table 3
Empirical performance of shortest-path and dynamic-programming heuristics for 64 randomly generated problems

Problem number	Number of			Share of choices			Performance ratio	
	Attributes	Levels	Consumers	OPT [c]	SP [a]	DP [b]	SP [a]	DP [b]
1	2	2	100	0.390	0.340	0.390	0.872	1.000
2	2	3	100	0.540	0.540	0.540	1.000	1.000
3	2	4	100	0.590	0.530	0.590	0.898	1.000
4	2	5	100	0.620	0.600	0.620	0.968	1.000
5	4	2	100	0.540	0.520	0.540	0.963	1.000
6	4	3	100	0.610	0.590	0.600	0.967	0.984
7	4	4	100	0.650	0.640	0.610	0.985	0.938
8	4	5	100	0.670	0.660	0.670	0.985	1.000
9	6	2	100	0.580	0.510	0.540	0.879	0.931
10	6	3	100	0.600	0.530	0.570	0.883	0.950
11	6	4	100	0.660	0.620	0.590	0.939	0.894
12	6	5	100	0.620	0.540	0.590	0.871	0.952
13	8	2	100	0.590	0.540	0.570	0.915	0.966
14	8	3	100	0.640	0.620	0.590	0.969	0.922
15	8	4	100	0.680	0.660	0.660	0.971	0.971
16	8	5	100	0.670	0.580	0.620	0.866	0.925
17	2	2	200	0.440	0.405	0.440	0.920	1.000
18	2	3	200	0.485	0.485	0.485	1.000	1.000
19	2	4	200	0.495	0.470	0.495	0.494	1.000
20	2	5	200	0.495	0.495	0.495	1.000	1.000
21	4	2	200	0.530	0.520	0.530	0.981	1.000
22	4	3	200	0.615	0.585	0.600	0.951	0.976
23	4	4	200	0.565	0.505	0.565	0.894	1.000
24	4	5	200	0.550	0.520	0.550	0.945	1.000
25	6	2	200	0.555	0.535	0.555	0.964	1.000
26	6	3	200	0.595	0.585	0.580	0.983	0.975
27	6	4	200	0.575	0.530	0.565	0.922	0.983
28	6	5	200	0.565	0.495	0.555	0.876	0.982
29	8	2	200	0.565	0.505	0.560	0.894	0.991
30	8	3	200	0.600	0.575	0.575	0.958	0.958
31	8	4	200	0.655	0.595	0.630	0.908	0.962
32	8	5	200	0.645	0.580	0.595	0.899	0.922
33	2	2	300	0.390	0.390	0.390	1.000	1.000
34	2	3	300	0.487	0.457	0.487	0.938	1.000
35	2	4	300	0.543	0.510	0.543	0.939	1.000
36	2	5	300	0.507	0.507	0.507	1.000	1.000
37	4	2	300	0.520	0.520	0.500	1.000	0.962
38	4	3	300	0.543	0.493	0.530	0.909	0.976
39	4	4	300	0.583	0.583	0.583	1.000	1.000
40	4	5	300	0.593	0.567	0.593	0.956	1.000
41	6	2	300	0.553	0.520	0.540	0.940	0.976
42	6	3	300	0.603	0.573	0.590	0.951	0.978
43	6	4	300	0.600	0.580	0.570	0.967	0.950
44	6	5	300	0.597	0.563	0.593	0.944	0.993
45	8	2	300	0.583	0.550	0.557	0.943	0.955
46	8	3	300	0.580	0.530	0.577	0.914	0.995
47	8	4	300	0.607	0.553	0.577	0.912	0.951
48	8	5	300	0.597	0.530	0.537	0.888	0.899
49	2	2	400	0.425	0.393	0.425	0.924	1.000
50	2	3	400	0.443	0.430	0.443	0.971	1.000
51	2	4	400	0.490	0.490	0.490	1.000	1.000
52	2	5	400	0.500	0.500	0.500	1.000	1.000
53	4	2	400	0.503	0.490	0.503	0.974	1.000
54	4	3	400	0.568	0.530	0.568	0.933	1.000
55	4	4	400	0.585	0.568	0.568	0.970	0.971
56	4	5	400	0.553	0.520	0.553	0.940	1.000

Table 3 (continued)

Problem number	Number of			Share of choices			Performance ratio	
	Attributes	Levels	Consumers	OPT [c]	SP [a]	DP [b]	SP [a]	DP [b]
57	6	2	400	0.533	0.528	0.533	0.990	1.000
58	6	3	400	0.558	0.518	0.538	0.927	1.000
59	6	4	400	0.573	0.560	0.555	0.977	0.969
60	6	5	400	0.578	0.523	0.573	0.904	0.991
61	8	2	400	0.563	0.555	0.558	0.986	0.991
62	8	3	400	0.565	0.518	0.518	0.916	0.917
63	8	4	400	0.595	0.568	0.563	0.954	0.946
64	8	5	400	0.593	0.525	0.563	0.885	0.949

[a] SP denotes the shortest-path heuristic.
[b] DP denotes the dynamic-programming heuristic.
[c] OPT denotes the optimal solution.

graphically displayed in Figure 1 for both heuristics. Observe that the distribution for the dynamic-programming heuristic strictly dominates the distribution for the shortest path heuristic. The worst performance of the dynamic-programming heuristic is 87.4% of the optimal, and the worst performance of the shortest-path heuristic is 86.6% of the optimal. On average, the dynamic-programming solution is 97.7% of the optimal solution, and the shortest-path solution is 94.4% of the optimal. These results suggest that both heuristics perform well with simulated data, the dynamic-programming heuristic somewhat better than the shortest-path heuristic.

6. Conclusion

The problem of maximizing the share of choices of a new, multiattribute product concept is shown to be NP-hard. A graph structure of the problem is used to develop an integer programming formulation of the problem, and to develop shortest-path and dynamic-programming heuristics. Both heuristics have arbitrarily-bad worst-case bounds. However, their empirical performance with simulated data is close to optimal, suggesting that they may be useful for solving practical problems. The dynamic-programming heuristic performs somewhat better than the shortest-path heuristic for the simulated problems.

The focus in this paper is on the problem of identifying a single multiattribute product. Selecting multiple items across which share is maximized is a more complex combinatorial problem that may be useful to investigate in future research. Although Green and Krieger (1985) propose heuristics to select product lines from a set of preselected items, solution procedures for the more difficult problem of constructing product lines directly from attribute levels still need to be developed.

The formulation presented in this paper assumes deterministic consumer choice. Alternative formulations in which preferences are related in a probabilistic manner to the utilities of choice-set items may be useful to investigate. Also, alternative objective functions, such as profit, may be incorporated in future models, although the practical usefulness of these formulations depends on the resolution of the problem of estimating multiattribute cost functions (Green, Carroll and Goldberg, 1981).

Appendix. Computational complexity of the shortest-path and dynamic-programming heuristics

Shortest-path heuristic

At stage k of the algorithm ($k = 2, 3, \ldots, K$), assume that the M largest elements are selected from S_{jk}. These elements are selected in $O[MJ_{k-1} + M \log M]$. The operations are repeated J_k times for each level of attribute k. Thus the complexity for a single stage is

$$J_k[MJ_{k-1} + M \log M].$$

For the K stages, the complexity is

$$\sum_{k=1}^{K} J_k[MJ_{k-1} + M \log M].$$

Let J be the average number of levels within an attribute. Setting $J_k = J_{k-1} = J$ in the above expression yields the total time complexity:

$$T(J, K, M) = O[JKM(J + \log M)].$$

For no infeasibilities (i.e., $M = 1$), this expression reduces to $T(J, K, M) = O[KJ^2]$.

Dynamic-programming heuristic

At stage k of the algorithm ($k = 1, 2, \ldots, K$), assume that M 'best' columns are carried forward into $S(k)$ from each of the $S_j(k)$ ($j = 1, 2, \ldots, J_k$). A total of MIJ_{k-1} comparisons are needed to compute the positive entries in each of the Mn_k, columns of $S_j(k)$. The selection of the 'best' M columns is accomplished in $O[IMJ_{k-1} + M \log M]$. These operations are repeated J_k times for each of the $S_j(k)$ matrices ($j = 1, 2, \ldots, J_k$). Thus the complexity for a single stage is

$$J_k[McJ_{k-1} + M \log M + IMJ_{k-1}],$$

where c is a constant. For the K stages, the complexity is

$$\sum_{k=1}^{K} J_k[McJ_{k-1} + M \log M + IMJ_{k-1}].$$

Let J be the average number of levels within an attribute. Setting $J_k = J_{k-1} = J$ in the above expression yields the total time complexity:

$$T(I, n, K, A) = O[KJM(JI + \log M)],$$

where $I \gg c$. For no infeasibilities (i.e., $M = 1$), this expression reduces to $T(I, J, K) = O[KIJ^2]$.

References

Addelman, S. (1962), "Orthogonal, main-effect plans for asymmetrical factorial experiments", *Technometrics* 4, 21–46.

Cattin, P., and Wittink, D.R. (1982), "Commercial use of conjoint analysis: A survey", *Journal of Marketing* 46, 44–53.

Cook, S.A. (1971), "The complexity of theorem-proving procedures", *Proceedings of the Third Annual ACM Symposium on the Theory of Computing*, 151–158.

Cornuejols, G., Fisher, M.L., and Nemhauser, G.L. (1977), "Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms", *Management Science* 23, 789–810.

Even, S., Itai, A., and Shamir, A. (1976), "On the complexity of timetable and multicommodity flow problems", *SIAM Journal of Computing* 5, 691–703.

Fisher, M.L. (1980), "Worst-case analysis of heuristic algorithm", *Management Science* 26, 1–17.

Garey, M.R., and Johnson, D.S. (1979), *Computers and Intractibility: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, CA.

Garey, M.R., Johnson, D.S., and Stockmeyer, L. (1976), "Some simplified NP-complete graph problems", *Theoretical Computer Science* 1, 237–267.

Green, P.E. (1984), "Hybrid models for conjoint analysis: An expository review", *Journal of Marketing Research* 2, 155–169.

Green, P.E., Carroll, J.D., and Goldberg, S.M. (1981), "A general approach to product design optimization via conjoint analysis", *Journal of Marketing* 45, 17–37.

Green, P.E., Carroll, J.D., Goldberg, S.M., and Kedia, P.K. (1981), "Product design optimization—A technical description of the POSSE methodology", Working Paper, University of Pennsylvania.

Green, P.E., and Krieger, A.M. (1984), "Models and heuristics for product line selection", *Marketing Science* 4, 1–19.

Green, P.E., and Srinivasan, V. (1978), "Conjoint analysis in consumer research: Issues and outlook", *Journal of Consumer Research* 5, 103–123.

Karp, R.M. (1975), "On the computational complexity of combinatorial problems", *Networks* 5, 45–68.

Kruskal, J.B. (1965), "Analysis of factorial experiments by estimating monotone transformations of data", *Journal of Royal Statistical Society*, Series B27, 251–263.

Papadimitriou, C.H., and Steiglitz, K. (1982), *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ.

Plackett, R.L., and Burman, J.P., (1946), "The design of optimum multifactor experiments", *Biometrika* 33, 305–325.

Srinivasan, V., and Shocker, A.D. (1973), "Estimating the weights for multiple attributes in a composite criterion using pairwise judgements", *Psychometrika* 38, 473–493.

Zufryden, F.S. (1977), "A conjoint measurement-based approach for optimal new product design and market segmentation", in: *Analytic Approaches to Product and Market Planning*, A.D. Shocker (ed.), Cambridge, MA: Marketing Science Institute, 100–114.