A Heuristic Approach to Product Design

Rajeev Kohli; Ramesh Krishnamurti

Management Science, Vol. 33, No. 12 (Dec., 1987), 1523-1533.

Stable URL:
http://links.jstor.org/sici?sici=0025-1909%28198712%2933%3A12%3C1523%3AAHATPD%3E2.0.CO%3B2-0

MANAGEMENT SCIENCE
Vol. 33, No. 12, December 1987
Printed in U.S.A.

A HEURISTIC APPROACH TO PRODUCT DESIGN*

RAJEEV KOHLI AND RAMESH KRISHNAMURTI
Joseph M. Katz Graduate School of Business, University of Pittsburgh, Pittsburgh, Pennsylvania 15260
School of Computing Science, Simon Fraser University, Burnaby, British Columbia, Canada V5A1S6

A dynamic-programming heuristic is described to find approximate solutions to the problem of identifying a new, multi-attribute product profile associated with the highest share-of-choices in a competitive market. The input data consist of idiosyncratic multi-attribute preference functions estimated using conjoint or hybrid-conjoint analysis. An individual is assumed to choose a new product profile if he/she associates a higher utility with it than with a status-quo alternative. Importance weights are assigned to individuals to account for differences in their purchase and/or usage rates and the performance of a new product profile is evaluated after taking into account its cannibalization of a seller's existing brands. In a simulation with real-sized problems, the proposed heuristic strictly dominates an alternative lagrangian-relaxation heuristic in terms of both computational time and approximation of the optimal solution. Across 192 simulated problems, the dynamic-programming heuristic identifies product profiles whose share-of-choices, on average, are 98.2% of the share-of-choices of the optimal product profile, suggesting that it closely approximates the optimal solution.
(MARKETING; PRODUCT DESIGN; CONJOINT ANALYSIS; HEURISTICS)

1. Introduction

Substantial effort has been devoted by marketing researchers to identifying and evaluating the performance of new product concepts prior to their market introduction. Consumer preference based methods such as conjoint analysis and multidimensional scaling have benefited the most for this interest. Improved data-collection methods have been proposed (e.g., Green 1984), and models and algorithms developed to evaluate the potential performance of new product concepts (e.g., Gavish, Horsky and Srikanth 1983; Green, Carroll and Goldberg 1981; Hauser and Simmie 1981).

This paper suggests a heuristic approach to new-product design using data from conjoint or hybrid-conjoint analysis. Specifically, it presents a dynamic-programming heuristic that provides an approximate solution to the problem of identifying a new, feasible, multi-attribute product profile associated with the highest share-of-choices in a competitive market. The proposed method assumes that individual preferences within a product class are described by a multi-attribute utility function, that all attributes have discrete levels, and that a new product is selected by an individual only if it has higher utility than a status-quo alternative.

When all attributes are discrete and preferences are idiosyncratic, conjoint choice-simulators (e.g., the QUALIN program of Green, Carroll and Goldberg's 1981 POSSE methodology) employ an enumeration procedure to identify a product profile with the highest share-of-choices. This presents no major computational difficulty if, as in the past, a small number of product profiles are tested. However, as commercial users have acquired more extensive experience with conjoint methodology, their demands have grown. Larger numbers of attributes and attribute-levels are being specified, especially when hybrid-conjoint models are used to collect preference data (Green 1984, p. 156). This in turn increases the number of product profiles evaluated and limits the use of

* Accepted by John R. Hauser; received June 5, 1986. This paper has been with the authors 4 months for 1 revision.

0025-1909/87/3312/1523$01.25

enumeration as a method for identifying the optimal product profile. For example, we describe an industry application (§5) in which a hybrid-conjoint model is used to collect data for a problem involving 16 attributes. A share-of-choices simulation for the 286-million product profiles is infeasible because it requires over 600 hours on a FPS-computer (which runs approximately 10 times faster than a DEC-10 computer and 2 times faster than a VAX-8600 computer). Even for a subproblem involving 10 attributes, simulating the share-of-choices for all product profiles takes over 1.5 hours on an FPS-computer.

The proposed heuristic, unlike explicit simulation, takes seconds to identify a product profile as its solution from a large set of feasible alternatives. For the preceding example, the heuristic is implemented in 2 and 1.2 seconds for the problems involving 16 and 10 attributes, respectively. Because it is computationally efficient and obtains solutions that are close to optimal, the proposed heuristic can also be used to perform sensitivity analysis (by resolving the problem for different subsets of respondents, and by perturbing utility-function estimates) and competitive analysis (by changing the set of competing products).

§2 provides a brief overview of preference based methods for new-product design. §3 describes, in the context of conjoint analysis, the problem of identifying a new product profile with the highest share-of-choices. §4 presents the proposed dynamic-programming heuristic, and §5 evaluates its performance.

2. Background

Preference based methods for new-product design have emphasized two methodologies: multidimensional scaling (MDS) and conjoint analysis. Each approach is described briefly below.

The Multidimensional Scaling Approach

The MDS approach describes consumer preferences in terms of brand and individual (ideal point) locations in an attribute space. In its simplest version, individual ideal-points are located in a common attribute space, the dimensions of which are differently weighted across individuals. A consumer is assumed to prefer products closer to the ideal point over those further away. Depending on the choice model employed, an individual may either be assumed to deterministically choose the brand closest to his/her ideal point or be seen to probabilistically select a brand where the probability of choice from a competing set is a decreasing function of the distance of the brand from the consumer's ideal point.

A number of researchers have presented procedures for identifying a point location in attribute-space that optimizes some objective function such as sales-revenue or share-of-choices received by a brand. Explicit solution procedures for maximizing the number of individuals for whom a new brand is closest to their respective ideal points are presented by Albers (1979), Albers and Brockhoff (1977, 1979), Gavish, Horsky and Srikanth (1983) and Zufryden (1979). Hauser and Simmie (1981) propose a model that employs a probabilistic choice-function, explicitly considers costs and prices, and includes a transformation from perceptual-space to the physical-dimensions of a product. However, they do not address the problem of cost measurement, which plagues all the proposed methodologies for product design optimization (Green, Carroll and Goldberg 1981). Also, none of the procedures deals with technological constraints on the variables' ranges. Computationally, all but the smallest problems take inordinately long to solve. Only Gavish, Horsky and Srikanth (1983) present efficient heuristics for solving problems of realistic sizes.

The Conjoint Analysis Approach

Conjoint analysis has been commercially used since the early 1970's (Green and Rao 1971; Johnson 1974). Unlike multidimensional scaling, this approach begins with the selection of a relevant set of attributes, each of which is described at a finite number of levels when the frequently-used part-worths model is employed. The attributes are used as factors in a fractional-factorial design, and the treatments of the design describe multi-attribute product profiles. Depending upon whether a conjoint or hybrid-conjoint model is employed, either all or some of these product profiles are evaluated by a respondent. In either case, at least some of the utility-function parameters (i.e., the part worths) are estimable at the individual level. Estimates of the part worths, obtained by scaling individual preferences (Green 1984; Srinivasan and Shocker 1973; Kruskal 1965), are then used to evaluate new product concepts (Green and Srinivasan 1978).

The first approach to product-design optimization using conjoint data was proposed by Zufryden (1977). He formulated the problem as a 0-1 integer-program that maximized the weighted share-of-choices for the new product. Each individual was assumed to deterministically choose the new product if she/he associated a higher utility with it than with a currently favored brand. Zufryden did not present any numerical examples of his approach, nor how the model could be implemented in terms of specific solution algorithms.

A comprehensive methodology called POSSE was introduced a few years later by Green, Carroll and Goldberg (1981). POSSE is a system of models and programs for carrying out a variety of steps in conjoint analysis ranging from stimulus-design to optimization and sensitivity-analysis. A unique feature of POSSE is its use of response-surface methods (Box, Hunter and Hunter 1978) to identify various objective functions that are subsequently employed in optimization routines. POSSE employs polynomial optimization when all attributes are continuous. For idiosyncratic preferences and all discrete attributes, it uses an enumeration to simulate the share-of-choices for new product profiles (Green, Carroll, Goldberg and Kedia 1981, p. 15). Combinations of discrete and continuous attributes are not directly considered in POSSE.

This paper addresses the problem of identifying a product profile that maximizes share-of-choices when individuals have different preferences over a set of multi-attribute alternatives and all attributes have discrete levels (combinations of fixed and continuous attributes are also not considered in this paper). In contrast to previous work on the problem, we emphasize a specific solution procedure that identifies "R" *independent*, feasible, product profiles as its solution, where "R" is a number specified by the user. The ability to identify more than one product profile is useful because a single, "optimal" product profile is seldom of interest in share-of-choices simulations. Rather, multiple product profiles are sought that are attractive in terms of their predicted shares-of-choices. The selected product profiles are evaluated in terms of their technological feasibility, their manufacturing and marketing costs, and their compatibility with the firm's current strategies and resources. One or more of these product profiles is then selected for further testing.

Like Zufryden (1977), we assume that an individual deterministically selects a brand with the highest utility in an offered set of alternatives. Individuals are assigned weights to account for differences in their purchase and/or usage rates, and the performance of a new product profile is evaluated after taking into account its cannibalization of a seller's existing-brand sales.

3. Description of Problem

First, let $\Omega = \{1, 2, \ldots, K\}$ denote the set of K attributes. For attribute $k \in \Omega$, let $\Phi_k = \{1, 2, \ldots, J_k\}$ denote the set of J_k levels. Further, let $\Theta = \{1, 2, \ldots, I\}$ denote the set

of I individuals. For individual i, let w_{ijk} denote the part worth of level j of attribute k. Let Θ_1 denote the subset of individuals in Θ whose currently favored (status-quo) brand is offered by a seller seeking to introduce a new brand, and let $\Theta_2 = \Theta - \Theta_1$ denote the subset of individuals for whom the status-quo brand is offered by a competitor. Let j_k^* denote the level of attribute k that appears in the product profile of the status-quo brand for individual i. Then

$$c_{ijk} = w_{ijk} - w_{ij_k^*k} \tag{1}$$

denotes the part worth of level j of attribute k *relative to* the part worth of level j_k^* of attribute k for individual i (we refer to the c_{ijk} as the "relative part worths" in the following discussion).[1]

Let u_{ip} denote the part-worths utility of product-profile p relative to the part-worths utility of the status-quo brand for individual i. Then u_{ip} has a value equal to the sum of the c_{ijk} across all levels of all attributes that appear in product-profile p. We assume that individual i chooses product-profile p over his/her status-quo brand only if $u_{ip} > 0$.

The share-of-choices for a test-profile is defined as the fraction of the number of individuals in Θ who choose it over their status-quo brand. Because the number of individuals in Θ is a constant, identifying a product profile p^* that maximizes the share-of-choices is equivalent to maximizing the number of individuals in Θ for whom $u_{ip^*} > 0$.

Cannibalization of the sales of brands currently offered by the seller is considered by maximizing over individuals in Θ_2 rather than Θ, because any increase in share obtained by the seller should result from customers switching from a competitor's brand (but not the seller's own brands) to the new product.[2]

Finally, let d_i denote the importance weight the seller assigns to individual $i \in \Theta_2$. Maximizing the share-of-choices for a new product profile (while controlling for cannibalization) is then equivalent to selecting a product profile p^* that maximizes the *weighted* number of individuals in Θ_2 for whom $u_{ip^*} > 0$, where d_i is the weight associated with individual i. The following heuristic is proposed to solve this problem.

4. 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 individual, each column to a distinct level of attribute k, and the ijth element to the relative part worth individual i associates with level j of attribute k. Let $C_j(k)$ denote the jth column of $C(k)$.

Step 1. To each column of $C(1)$, add column $C_j(2), j \in \Phi_2$. Call the resulting matrix $S_j(2)$; i.e.,

$$S_j(2) = C(1) + [C_j(2)][1] \qquad \text{for all} \qquad j \in \Phi_2 \tag{2}$$

where [1] denotes a conformable row-vector of unit elements.[3] Observe that there are as

[1] This discussion assumes a main-effects model of individual utilities and can be extended in an obvious manner to include interaction effects.

[2] Note that this controls for cannibalization of the seller's sales, but not profits.

[3] For example, if there are two individuals in Θ_2 ($i = 1, 2$) and attribute 1 has two levels ($j = 1, 2$), then

$$C_1(1) = (c_{11}(1)c_{21}(1))'; \qquad C_2(1) = (c_{12}(1)c_{22}(1))';$$

$$C_j(2) = (c_{1j}(2)c_{2j}(2))' \qquad \text{for all} \qquad j \in J_2; \qquad \text{and}$$

$$S(j2) = \begin{bmatrix} c_{11}(1) & c_{12}(1) \\ c_{21}(1) & c_{22}(1) \end{bmatrix} + \begin{bmatrix} c_{1j}(2) \\ c_{2j}(2) \end{bmatrix} [1 \quad 1]$$

$$= \begin{bmatrix} c_{11}(1) + c_{1j}(2) & c_{12}(1) + c_{1j}(2) \\ c_{21}(1) + c_{1j}(2) & c_{22}(1) + c_{2j}(2) \end{bmatrix}.$$

many matrices $S_j(2)$ as there are levels of attribute 2. For each level j of attribute 2, select the column from $S_j(2)$ that has the highest (weighted) number of positive elements, where d_i is the weight for the ith element (individual). Ties are broken by (a) selecting the tied column that has the highest (weighted) number of nonnegative elements, and (b) selecting from still-tied columns the one with the highest (weighted) *sum* of nonnegative elements. If any columns are still tied, one of them is randomly selected.

Let the selected column from $S_j(2)$ be denoted $s_j^*(2)$ for all $j \in \Phi_2$. Form the matrix

$$S^*(2) = [s_1^*(2)s_2^*(2)\cdots s_{J_2}^*(2)]. \tag{3}$$

For each individual $i \in \Theta_2$, the value of $s_{ij}^*(2)$ (the ijth element of $S^*(2)$) is the sum of the relative part worths for level j of attribute 2 and level j_1 of attribute 1, where j_1 is chosen to maximize the (weighted) number of individuals in Θ_2 for whom the sum $c_{ij_1}(1) + c_{ij}(2)$ is positive.

Step 2. To each column of $S^*(2)$, add column $C_j(3), j \in \Phi_3$. Call the resulting matrix $S_j(3)$; i.e.,

$$S_j(3) = S^*(2) + [C_j(3)][1] \qquad \text{for all} \qquad j \in \Phi_3, \tag{4}$$

where [1] denotes a conformable row-vector of unit elements. For each level j of attribute 3, select the column from $S_j(3)$ that has the highest (weighted) number of positive elements. Ties are broken as described in Step 1 above. Let the selected column from $S_j(3)$ be denoted $s_j^*(3)$ for all $j \in \Phi_3$. Form the matrix

$$S^*(3) = [s_1^*(3)s_2^*(3)\cdots s_{J_3}^*(3)]. \tag{5}$$

For each individual $i \in \Theta_2$, the value of $s_{ij}^*(3)$ is the sum of the part worths for level j of attribute 3, level j_2 of attribute 2, and level j_1 of attribute 1. Given level j of attribute 3, level j_2 of attribute 2 is chosen to maximize the (weighted) number of individuals for whom the sum

$$s_{ij_2}^*(2) + c_{ij}(3) \tag{6}$$

is positive; and given level j_2 of attribute 2, level j_1 of attribute 1 is selected as in Step 1.

Step $(k-1)$ (*General step*). To each column of $S^*(k-1)$, add column $C_j(k)$ of matrix $C(k)$. Call the resulting matrix $S_j(k)$; i.e.,

$$S_j(k) = S^*(k-1) + [C_j(k)][1] \qquad \text{for all} \qquad j \in \Phi_k, \tag{7}$$

where [1] denotes a conformable row-vector of unit elements. For each level j of attribute k, select the column from $S_j(k)$ that has the highest (weighted) number of positive elements. Ties are broken as described in Step 1.

Let the selected column from $S_j(k)$ be denoted $s_j^*(k)$ for all $j \in \Phi_k$. Form the matrix

$$S^*(k) = [s_1^*(k)s_2^*(k)\cdots s_{J_k}^*(k)]. \tag{8}$$

For each individual $i \in \Theta_2$, the value of $s_{ij}^*(k)$ is the sum of the part worths for level j of attribute k and one level of each of attributes $1, 2, \ldots, (k-1)$. Given level j of attribute k, level j_{k-1} of attribute $k-1$ is chosen to maximize the number of individuals for whom the sum

$$s_{ij_{k-1}}^*(k-1) + c_{ij}(k) \tag{9}$$

is positive; given level j_{k-1} of attribute $k-1$, level j_{k-2} of attribute $k-2$ is selected to maximize the number of individuals for whom the sum

$$s_{ij_{k-2}}^*(k-2) + c_{ij_{k-1}}(k-1) \tag{10}$$

is positive; and so on down to attribute 2, where for each level j_2 of attribute 2, level j_1 of attribute 1 is selected as described in Step 1.

We emphasize that the proposed algorithm is a *heuristic* because for each level of an

attribute only a level of a *preceding* (and no other) attribute is selected. The solution is identified by the column that contains the largest (weighted) number of positive elements in $S^*(K)$. The ith element of this column represents the ith individual's part-worths utility for the selected product profile.

Infeasible Solutions

The above heuristic identifies a single product profile as the solution. However, in practice, certain product profiles may be infeasible due to technological and/or cost constraints, and a user may wish to examine more than one feasible solution-profile. Therefore a procedure is needed to ensure that the product profile identified is not infeasible, and also to permit a user to specify the number of feasible product profiles identified by the heuristic.

Let N denote the number of infeasible product profiles. Let level j of attribute k appear in N_{jk} ($\leq N$) infeasible product-profiles. Let N_k be the maximum of the number of infeasible product-profiles across the J_k levels of attribute k; i.e.,

$$N_k = \max \{N_{jk} | j \in \Phi_j\}. \tag{11}$$

Let $N_{\min} = \min \{N_k | k \in \Omega\}$ 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, the dynamic-programming heuristic is altered in two ways. First, the algorithm is implemented with attribute k' (for which $N_{k'} = N_{\min}$) considered *last* in the algorithm. Second, each $S^*(k)$ is constructed by selecting M ($>N_{\min}$) columns, instead of 1 column, from each $S_j(k)$, provided the $S_j(k)$ have M columns each; otherwise, all columns of $S_j(k)$ are carried over into the $S^*(k)$ matrix. 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 $M = N_{\min} + R$.

5. Performance Evaluation

The predicted shares-of-choices for product profiles identified by the dynamic-programming heuristic are compared to the predicted shares-of-choices for (a) product profiles identified by an alternative lagrangian-relaxation heuristic, and (b) corresponding optimal product profiles.[4] Also, the computational times for the dynamic-programming heuristic are compared to those for the lagrangian-relaxation heuristic and for an enumeration procedure for identifying optimal product profiles.

Forty-eight problems were generated using a 3×4^2 experimental design that employed number-of-individuals (100, 200, 300, 400), number-of-attributes (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 (e.g., over 60 hours on an FPS-computer for a problem involving 400 individuals and 10 attributes at 5 levels each). All part worths were randomly selected from a uniform distribution on [0, 1], then normalized within

[4] Details of the lagrangian-relaxation heuristic are available in an appendix that can be obtained from TIMS.

individuals. Idiosyncratic status-quo product profiles were randomly chosen, and a unit importance-weight was assigned to each individual. All product profiles were assumed to be feasible.

Each problem was solved using the dynamic-programming and lagrangian-relaxation heuristics, and also via an enumeration of all product profiles. For the 48 problems, the dynamic-programming heuristic strictly dominates the lagrangian-relaxation heuristic in terms of both computational time and approximation of the optimal solution. Table 1 (column 12, problem #13) shows that the share-of-choices for the lagrangian-relax-

TABLE 1

Relative Performance of Dynamic-Programming and Lagrangian-Relaxation Heuristics for 48 *Simulated Problems*

Problem Number	No. of Individuals	No. of Attributes	No. of Levels per Attribute	n(DP)*	n(LR)**	n(OPT)***	t(DP)#	t(LR)##	t(ENUM)###	n(DP) ÷ n(OPT)	n(LR) ÷ n(OPT)	t(DP) ÷ t(ENUM)	t(LR) ÷ t(ENUM)
1	100	4	2	57	47	57	0.025	1.855	0.054	1.00	0.82	0.4630	34.3519
2	200	4	2	108	105	108	0.049	4.430	0.107	1.00	0.97	0.4579	41.4019
3	300	4	2	170	156	170	0.073	5.430	0.160	1.00	0.92	0.4563	33.9375
4	400	4	2	223	191	223	0.096	8.245	0.213	1.00	0.86	0.4507	38.7089
5	100	4	3	57	39	57	0.049	5.652	0.247	1.00	0.68	0.1984	22.8826
6	200	4	3	108	99	108	0.096	6.320	0.492	1.00	0.92	0.1951	12.8455
7	300	4	3	164	154	164	0.144	10.355	0.737	1.00	0.94	0.1954	14.0502
8	400	4	3	235	200	235	0.192	15.420	0.984	1.00	0.85	0.1951	15.6707
9	100	4	4	68	46	68	0.083	3.430	0.766	1.00	0.68	0.1084	4.4778
10	200	4	4	107	96	107	0.161	8.850	1.524	1.00	0.90	0.1056	5.8071
11	300	4	4	165	151	167	0.241	16.585	2.285	0.99	0.90	0.1055	7.2582
12	400	4	4	227	203	227	0.321	24.035	3.046	1.00	0.89	0.1054	7.8907
13	100	4	5	62	36	62	0.124	4.940	1.855	1.00	0.58	0.0668	2.6631
14	200	4	5	113	94	116	0.244	12.565	3.702	0.97	0.81	0.0659	3.3941
15	300	4	5	170	132	170	0.364	22.485	5.545	1.00	0.78	0.0656	4.0550
16	400	4	5	208	195	217	0.483	37.600	7.384	0.96	0.90	0.0654	5.0921
17	100	6	2	58	52	58	0.040	2.370	0.249	1.00	0.90	0.1606	9.5181
18	200	6	2	110	102	110	0.077	4.900	0.496	1.00	0.93	0.1552	9.8790
19	300	6	2	163	163	165	0.115	8.315	0.744	0.99	0.99	0.1546	11.1761
20	400	6	2	220	200	220	0.153	12.270	0.992	1.00	0.91	0.1542	12.3690
21	100	6	3	64	46	67	0.080	3.945	2.749	0.96	0.69	0.0291	1.4351
22	200	6	3	109	101	113	0.156	9.285	5.477	0.96	0.89	0.0285	1.6953
23	300	6	3	162	146	168	0.233	18.135	8.214	0.96	0.87	0.0284	2.2078
24	400	6	3	228	196	228	0.311	21.995	10.958	1.00	0.86	0.0284	2.0072
25	100	6	4	65	47	65	0.135	13.195	15.390	1.00	0.72	0.0088	0.8574
26	200	6	4	111	98	115	0.264	15.750	30.678	0.97	0.85	0.0086	0.5134
27	300	6	4	171	153	176	0.395	27.110	46.057	0.97	0.87	0.0086	0.5886
28	400	6	4	209	204	213	0.522	44.580	61.281	0.98	0.96	0.0085	0.7275
29	100	6	5	68	49	69	0.205	16.420	58.726	0.99	0.71	0.0035	0.2796
30	200	6	5	124	101	124	0.402	22.990	117.041	1.00	0.81	0.0034	0.1964
31	300	6	5	176	133	177	0.601	44.200	175.538	0.99	0.75	0.0034	0.2518
32	400	6	5	234	187	237	0.799	76.155	233.999	0.99	0.79	0.0034	0.3255
33	100	8	2	56	43	56	0.054	3.105	1.176	1.00	0.77	0.0459	2.6403
34	200	8	2	120	101	120	0.106	6.580	2.351	1.00	0.84	0.0451	2.7988
35	300	8	2	171	159	171	0.157	11.560	3.522	1.00	0.93	0.0446	3.2822
36	400	8	2	228	211	230	0.209	15.970	4.698	0.99	0.92	0.0445	3.3993
37	100	8	3	69	51	69	0.111	10.030	29.913	1.00	0.74	0.0037	0.3353
38	200	8	3	110	88	121	0.216	17.265	59.683	0.91	0.73	0.0036	0.2893
39	300	8	3	171	142	173	0.323	32.430	89.484	0.99	0.82	0.0036	0.3624
40	400	8	3	225	195	225	0.429	38.080	119.229	1.00	0.87	0.0036	0.3194
41	100	8	4	61	42	65	0.187	13.310	298.345	0.94	0.65	0.0006	0.0446
42	200	8	4	122	99	126	0.368	29.040	596.004	0.97	0.79	0.0006	0.0487
43	300	8	4	172	148	176	0.548	45.050	893.404	0.98	0.84	0.0006	0.0504
44	400	8	4	224	202	234	0.730	65.250	1191.156	0.96	0.86	0.0006	0.0548
45	100	8	5	65	42	67	0.283	21.820	1776.824	0.97	0.63	0.0002	0.0123
46	200	8	5	112	91	118	0.556	31.860	3548.414	0.95	0.77	0.0002	0.0090
47	300	8	5	176	143	185	0.835	83.780	5323.910	0.95	0.77	0.0002	0.0157
48	400	8	5	230	179	233	1.108	124.455	7094.351	0.99	0.77	0.0002	0.0175

* n(DP) denotes the number of individuals for whom the dynamic-programming product profile has higher utility than the status-quo product profile.
** n(LR) denotes the number of individuals for whom the lagrangian-relaxation product profile has higher utility than the status-quo product profile.
*** n(OPT) denotes the number of individuals for whom the optimal product profile has higher utility than the status-quo product profile.
# t(DP) denotes the computational-time (in seconds) for the dynamic-programming heuristic.
t(LR) denotes the computational-time (in seconds) for the lagrangian-relaxation heuristic.
t(ENUM) denotes the computational-time (in seconds) for enumeration.

ation product profile is always higher than 58% of the share-of-choices for the optimal product profile. In contrast, the share-of-choices for the dynamic-programming-product-profile (column 11, problem #38) is always higher than 91% of the share-of-choices for the optimal-product-profile. Also, the gap between the computational times for the two heuristics increases with problem sizes: for the smallest problem (#1) the dynamic-programming heuristic is 74 times faster, and for the largest problem (#48) it is 100 times faster than the lagrangian-relaxation heuristic. The dynamic-programming heuristic also dominates the enumeration procedure computationally for all 48 problems. In contrast, the lagrangian-relaxation heuristic dominates the enumeration for 21 of the 48 problems (#24–#32, and #37–#48).

TABLE 2

Performance of Dynamic-Programming Heuristic for 192 *Randomly Generated Problems*

Problem Number	No. of Individuals	No. of Attributes	No. of Levels per Attribute	n(DP)* Replicate				n(OPT)** Replicate				n(DP) ÷ n(OPT) Replicate				Average n(DP) ÷ n(OPT)
				1	2	3	4	1	2	3	4	1	2	3	4	
1	100	4	2	57	53	56	57	57	53	56	57	1.00	1.00	1.00	1.00	1.00
2	200	4	2	108	113	106	113	108	113	106	113	1.00	1.00	1.00	1.00	1.00
3	300	4	2	170	161	153	159	170	161	153	159	1.00	1.00	1.00	1.00	1.00
4	400	4	2	223	208	212	215	223	208	212	215	1.00	1.00	1.00	1.00	1.00
5	100	4	3	57	60	59	52	57	60	59	54	1.00	1.00	1.00	0.96	0.99
6	200	4	3	108	114	111	113	108	118	111	116	1.00	0.97	1.00	0.97	0.98
7	300	4	3	164	160	174	164	164	160	174	164	1.00	1.00	1.00	1.00	1.00
8	400	4	3	235	221	201	220	235	221	202	230	1.00	1.00	1.00	0.96	0.99
9	100	4	4	68	65	69	59	68	65	69	59	1.00	1.00	1.00	1.00	1.00
10	200	4	4	107	111	105	115	107	113	110	115	1.00	0.98	0.95	1.00	0.98
11	300	4	4	165	164	167	162	167	164	167	162	0.99	1.00	1.00	1.00	1.00
12	400	4	4	227	235	219	220	227	235	219	220	1.00	1.00	1.00	1.00	1.00
13	100	4	5	58	56	62	58	58	58	64	59	1.00	0.97	0.97	0.98	0.98
14	200	4	5	113	109	127	110	116	109	127	110	0.97	1.00	1.00	1.00	0.99
15	300	4	5	170	171	176	165	170	171	176	166	1.00	1.00	1.00	0.99	1.00
16	400	4	5	208	227	209	226	217	227	210	226	0.96	1.00	1.00	1.00	0.99
17	100	6	2	58	58	52	59	58	61	57	59	1.00	0.95	0.91	1.00	0.96
18	200	6	2	110	99	113	109	110	100	113	113	1.00	0.99	1.00	0.96	0.99
19	300	6	2	163	158	161	160	165	160	161	160	0.99	0.99	1.00	1.00	0.99
20	400	6	2	220	211	223	216	220	211	223	216	1.00	1.00	1.00	1.00	1.00
21	100	6	3	64	61	59	63	67	64	61	65	0.96	0.95	0.97	0.95	0.96
22	200	6	3	109	113	121	115	113	113	123	119	0.96	1.00	0.98	0.97	0.98
23	300	6	3	162	162	165	165	168	168	169	172	0.96	0.96	0.98	0.96	0.96
24	400	6	3	228	214	213	233	228	223	215	233	1.00	0.96	0.99	1.00	0.99
25	100	6	4	65	55	68	64	65	57	69	65	1.00	0.96	0.99	0.98	0.98
26	200	6	4	111	115	120	113	115	116	121	118	0.97	0.99	0.99	0.96	0.98
27	300	6	4	171	172	161	177	176	175	172	177	0.97	0.98	0.94	1.00	0.97
28	400	6	4	209	242	221	219	213	242	230	219	0.98	1.00	0.96	1.00	0.98
29	100	6	5	68	58	65	64	69	61	66	66	0.99	0.95	0.98	0.97	0.97
30	200	6	5	124	124	121	119	124	126	123	124	1.00	0.98	0.98	0.96	0.98
31	300	6	5	176	174	175	176	177	176	176	176	0.99	0.99	0.99	1.00	0.99
32	400	6	5	234	222	227	216	237	227	228	219	0.99	0.98	1.00	0.99	0.99
33	100	8	2	56	64	66	56	56	64	66	57	1.00	1.00	1.00	0.98	0.99
34	200	8	2	120	113	112	105	120	119	113	108	1.00	0.95	0.99	0.97	0.98
35	300	8	2	171	167	164	163	171	169	167	163	1.00	0.99	0.98	1.00	0.99
36	400	8	2	228	212	221	223	230	218	223	224	0.99	0.97	0.99	1.00	0.99
37	100	8	3	69	63	65	56	69	64	66	57	1.00	0.98	0.98	0.98	0.98
38	200	8	3	110	115	129	111	121	115	129	113	0.91	1.00	1.00	0.98	0.97
39	300	8	3	171	176	171	167	173	177	171	174	0.99	0.99	1.00	0.96	0.98
40	400	8	3	225	230	217	229	225	236	221	232	1.00	0.97	0.98	0.99	0.99
41	100	8	4	61	62	62	63	65	63	65	65	0.94	0.98	0.95	0.97	0.96
42	200	8	4	122	122	128	115	126	127	128	123	0.97	0.96	1.00	0.93	0.96
43	300	8	4	172	169	177	174	176	173	179	179	0.98	0.98	0.99	0.97	0.98
44	400	8	4	224	219	233	225	234	226	234	233	0.96	0.97	1.00	0.97	0.97
45	100	8	5	65	65	63	63	67	69	68	69	0.97	0.94	0.93	0.91	0.94
46	200	8	5	112	118	122	121	118	125	128	121	0.95	0.94	0.95	1.00	0.96
47	300	8	5	176	168	175	165	185	179	183	175	0.95	0.94	0.96	0.94	0.95
48	400	8	5	230	226	229	230	233	237	236	230	0.99	0.95	0.97	1.00	0.98

* n(DP) denotes the number of individuals for whom the dynamic-programming product profile has higher utility than the status-quo product profile.

** n(OPT) denotes the number of individuals for whom the optimal product profile has higher utility than the status-quo product profile.

To further test the performance of the dynamic-programming heuristic, optimal and dynamic-programming share-of-choices were compared for an additional 144 problems. These problems were also generated using the previously described experimental design, three more randomly-generated problems being solved for each of the 48 distinct combinations of the number-of-individuals, number-of-attributes, and number-of-levels-per-attribute. Including the previous 48 problems, a total of 192 problems were solved using the dynamic-programming heuristic and enumeration.

Table 2 presents the results for the 192 problems (computational times for the dynamic-programming and enumeration procedures are the same for each problem-type as described in Table 1). The shares-of-choices of the dynamic-programming product profiles are never below 91%, and on average are 98.2%, of the shares-of-choices of the optimal product profiles. The optimal product profile is identified by the heuristic in 46.35% (89 out of 192) cases.

Finally, the dynamic-programming heuristic was implemented for an actual dataset, obtained from a hybrid-conjoint study for a consumer-durable product. One hundred and eighty seven individuals participated in the study. Sixteen attributes were employed, 8 attributes appearing at 4 levels each, 7 attributes at 3 levels each, and 1 attribute at 2 levels. Idiosyncratic part worths were estimated, and a status-quo product profile specified for each respondent. The dynamic-programming heuristic identified a solution product-profile in 2 seconds. However, this solution could not be compared to the optimal solution because an enumeration of all product profiles requires over 600 hours on an FPS-computer. The heuristic and optimal solutions were therefore compared for a subproblem involving 10 attributes that can be solved in reasonable time by the enumeration. The 16 attributes were ranked in descending order of their average-part-worth ranges and the 10 highest-ranked attributes selected for the subproblem. The dynamic-programming heuristic was implemented in 1.2 seconds, the enumeration in 1.5 hours. Both methods identified the same product profile. Therefore for at least this problem the proposed heuristic performed as well as the enumeration procedure in terms of the solution obtained.

These results suggest that the dynamic-programming heuristic should perform well in practice. However in certain instances the heuristic can identify a product profile whose predicted share-of-choices is not close to the share-of-choices of the optimal product profile. The situations in which the dynamic-programming heuristic performs poorly is illustrated by an example involving 3 attributes, one at 3 levels and the other two at 2 levels each. Consider the 2 segments below with identical part worths for respondents within (but not between) segments:

	Attribute						
	1			2		3	
	Level			Level		Level	
Segment	1	2	3	1	2	1	2
1 (n_1 individuals)	2δ	$(-1/2) - \delta$	0	$-\delta$	0	$(1/2) + 2\delta$	0
2 (n_2 individuals)	$-1 + \delta$	$(-1/2) - \delta$	0	$-\delta$	0	$(1/2) + 2\delta$	0

Let δ be a small, positive number, and let $n_2 > n_1$. The dynamic-programming heuristic selects the product profile with levels 1, 2 and 1 of attributes 1, 2 and 3, respectively. Only individuals in segment 1 prefer this product profile over their status-quo product profile. The optimal product profile has levels 2, 2 and 1 of attributes 1, 2 and 3, respectively. Only individuals in segment 2 prefer this product profile to their status-quo product profile. Therefore the ratio of the share-of-choices for the dynamic-programming-product-profile and the optimal-product-profile is n_1/n_2. Because $n_2 > n_1$,

this ratio is less than 1, and the smaller the ratio, the worse the performance of the dynamic-programming heuristic.[5]

The dynamic-programming heuristic performs poorly in this case because of two reasons. First, both segments have the same relative part-worths except for level 1 of attribute 1, which individuals in segment 1 marginally prefer to the status-quo level but individuals in segment 2 prefer significantly less than the status-quo level. Second, attribute 1 is followed by an attribute (i.e., attribute 2) that is unimportant in the sense that its relative part worths for both segments are either zero (level 2) or $-\delta$ (level 1). Consequently, attribute 2 does not contribute in a significant manner to the sum of the part worths, but eliminates level 1 of attribute 1 from subsequent consideration because, for both of its levels, level 1 of attribute 1 is selected. However, it can be verified that for any other ordering of attributes, the dynamic-programming heuristic recovers the optimal product profile. This suggests that it is preferable to perform multiple runs of the proposed heuristic for a given problem, different orderings of the attributes being employed in different runs. Also, the analysis should preferably be performed at a segment, rather than market, level so that preferences of respondents are not widely divergent (as they are for level 1 of attribute 1 in the above example).

6. Conclusion

A heuristic procedure is presented to find approximate solutions to the problem of identifying a new, feasible multi-attribute product profile associated with the highest share-of-choices in a competitive market. The input data are comprised of idiosyncratic part-worth utilities, estimated using conjoint or hybrid-conjoint analysis. Different importance weights are assigned to individuals and the cannibalization of existing brand-sales is controlled. A simulation indicates that the proposed heuristic dominates an alternative lagrangian-relaxation heuristic in terms of both computational time and approximation of the optimal solution. Computationally, the dynamic-programming heuristic is significantly more efficient than an enumeration procedure for simulating shares-of-choices, and also identifies product profiles whose share-of-choices is optimal or near-optimal.[6]

[5] If $n_2 \gg n_1$, the ratio n_1/n_2 approaches zero, resulting in an arbitrarily-bad performance of the dynamic-programming heuristic.

[6] The authors express their appreciation to Finbar McEvoy and R. Sukumar for computer programming assistance. Thanks are also expressed to Paul E. Green for kindly furnishing the data for the industry application. This research was supported by a Faculty Research Grant from the Joseph M. Katz Graduate School of Business, University of Pittsburgh.

References

ALBERS, S., "An Extended Algorithm for Optimal Product Positioning," *European J. Oper. Res.*, 3 (May 1979), 222–231.

——— AND K. BROCKHOFF, "A Procedure for New Product Positioning in Attribute Space," *European J. Oper. Res.*, 1 (January 1977), 230–238.

BOX, GEORGE E. P., WILLIAM G. HUNTER AND J. STUART HUNTER, *Statistics for Experimenters*, John Wiley and Sons, New York, 1978.

GAVISH, B., D. HORSKY AND K. SRIKANTH, "An Approach to the Optimal Positioning of a New Product," *Management Sci.*, 29, 11 (November 1983), 1277–1297.

GEOFFRION, ARTHUR M., "Lagrangian Relaxation for Integer Programming," *Math. Programming Study*, 2 (1974), 82–114.

GREEN, PAUL E., "Hybrid Models for Conjoint Analysis: An Expository Review," *J. Marketing Res.*, 21 (May 1984), 155–169.

———, J. DOUGLAS CARROLL AND STEPHEN M. GOLDBERG, "A General Approach to Product Design Optimization via Conjoint Analysis," *J. Marketing*, 45 (1981), 17–37.

———, ———, ——— AND PRADEEP K. KEDIA, "Product Design Optimization—A Technical Description of the POSSE Methodology," Working paper, University of Pennsylvania, September 1981.

——— AND VITHALA R. RAO, "Conjoint Measurement for Quantifying Judgemental Data," *J. Marketing Res.*, (August 1971), 353–363.

——— AND V. SRINIVASAN, "Conjoint Analysis in Consumer Research: Issues and Outlook," *J. Consumer Res.*, 5 (September 1978), 103–123.

HAUSER, JOHN R. AND PATRICIA SIMMIE, "Profit Maximizing Perceptual Positions: An Integrated Theory for the Selection of Product Features and Price," *Management Sci.*, 27 (January 1981), 33–56.

JOHNSON, RICHARD M., "Trade-off Analysis of Consumer Values," *J. Marketing Res.*, (May 1974), 121–127.

KRUSKAL, JOSEPH B., "Analysis of Factorial Experiments by Estimating Monotone Transformations of Data," *J. Roy. Statist. Soc., Ser.* B, 27 (1965), 251–263.

PAPADIMITRIOU, C. H. AND K. STIGLITZ, *Combinatorial Optimization: Algorithms and Complexity*, Prentice Hall, Englewood Cliffs, N.J., 1982.

SRINIVASAN V. AND ALLAN D. SHOCKER, "Estimating the Weights for Multiple Attributes in a Composite Criterion Using Pairwise Judgements," *Psychometrika*, 38 (December 1973), 473–493.

ZUFRYDEN, FRED S., "A Conjoint Measurement-Based Approach for Optimal New Product Design and Market Segmentation," in *Analytic Approaches to Product and Market Planning*, Alan D. Shocker (Ed.), Marketing Science Institute, Cambridge, MA, 1977, 100–114.

———, "ZIPMAP—A Zero-One Integer Programming Model for Market Segmentation and Product Positioning," *J. Oper. Res. Soc.*, 30 (1979), 63–70.