European Journal of Operational Research 128 (2001) 58–73

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/dsw

Theory and Methodology

A pruning heuristic for use with multisource product design

Peter Tarasewich *, Patrick R. McMullen ☆

Maine Business School, University of Maine, 5723 Corbett Business Building, Orono, ME 04469-5723, USA

Received 3 June 1998; accepted 19 May 1999

Abstract

Products can be improved by integrating multiple viewpoints during the design process. A model has been developed that uses conjoint data from consumers and designers and optimizes a product design based on the total share-of-choices. Because the problem becomes very difficult to solve as size increases, a heuristic is developed, based on pruning techniques, to solve the problem to near-optimality in a shorter period of time as compared to complete enumeration. The performance of the heuristic is demonstrated through the use of test data and by comparison to a Genetic Algorithm (GA) based heuristic and Tabu search. Structural results for the heuristic are also provided. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Genetic algorithms; Tabu search; Heuristics; Product design; Conjoint analysis

1. Introduction

Increasing emphasis is being placed on product design as a strategic business activity. Conjoint analysis (CA) procedures have become a very popular way of collecting consumer data to be used in the product design process. Products can be defined as consisting of various attributes (e.g. engine size) each with two or more levels (e.g. 6, 9, and 12 horsepower). A product design consists of choosing one level from each of the different attributes. Subjects order a subset of possible product designs from most favored to least favored, making tradeoffs between the various levels of the different attributes. This information is then decomposed into a set of part worths (component utilities) for each level of each product attribute for each subject. The utility values can be used to determine each subject's preference for *any* possible product design. When all subjects are taken together as a group, an optimal product design can be determined. An optimal product design is defined as the design that optimizes some criteria, such as maximizing the number of consumers

* Correspoding author. Tel.: +1-207-581-1995; fax: +1-207-581-1956.
E-mail address: tarase@maine.edu (P. Tarasewich).
☆ *Current address*: Department of Management, College of Business, Auburn University, Auburn, AL 36849, USA.

PII: S 0 3 7 7 - 2 2 1 7 (9 9) 0 0 3 5 0 - 1

who will choose, or prefer, that product design to a status-quo (current) design. This can also be thought of as maximizing share-of-choices, or, in other words, market share.

Many methods have been developed to take conjoint data and produce optimal or near-optimal product designs. Kohli and Krishnamurti (1987) developed a dynamic-programming-like heuristic to find solutions to the problem of identifying a new, multiattribute product profile associated with the highest share-of-choices in a competitive market. This heuristic was then extended to cover product line design (Kohli and Sukumar, 1990). Nair et al. (1995) developed improved heuristics using beam-search methods to more effectively solve this problem. Balakrishnan and Jacob (1996) used Genetic Algorithms, which are based on population genetics ("survival of the fittest"), with conjoint data to generate product designs that are near optimal. Heuristic approaches are justified because the product design problem has been shown to be very difficult (NP-hard) to solve to optimality in a reasonable amount of time (Kohli and Krishnamurti, 1989).

But each of these methods, while significantly contributing to the product design process, still has limitations. Most significantly, conjoint analysis designs a product based only on input from the *consumer*. We argued (Tarasewich and Nair, 1999) that designer preferences are as important as consumer preferences in the product design problem, and created a method to produce an optimal product design based on distinct and parallel opinions from the consumer and the *designer*. Consumer attributes, designer characteristics, and the relationship between them are defined. Conjoint analysis is used to collect data from the two parties. Optimization of a product design is then achieved by maximizing the share-of-choices for both consumers and designers.

Since the product design problem using multisource data is very difficult to solve, efficient methods are needed. This paper introduces a heuristic, based on pruning techniques, that is used to solve the problem. The heuristic is compared to other solution procedures – a Genetic Algorithm based heuristic and Tabu search. The pruning heuristic compares favorably to these other solution methods.

This paper is organized as follows. Section 2 briefly explains the need for integrating designer and consumer perspectives. Section 3 reviews the share-of-choices model for multisource product design. Section 4 introduces a pruning heuristic used to solve our problem. Section 5 presents computational results and comparisons against other solution methods. Section 6 provides some structural results, and closing remarks are given in Section 7.

2. The need for integrating consumer and designer preferences

Previous research (Tarasewich and Nair, 1999) has already addressed in detail the need for integrating multiparty preferences into product designs. Often times consumer preferences are used exclusively to design products. But designer preferences can be an important source of expertise that needs to supplement consumer preferences in the product design problem. Designs made solely using consumer preferences may be unrealistic. Designers acting alone may come up with a product they think is a technological marvel, but that consumers see as silly and unusable. Taking only one set of preferences into account may not result in the best possible product in the long term. For example, consumers may desire a "user-friendly" personal computer that is easy to begin using. But the designer may realize that the computer should also be designed to meet the users' longer-term needs. Designers should have the freedom to create innovative product designs that not only meet current user requirements but also predicted future consumer expectations.

The way that a consumer looks at product attributes is usually much different from the way a designer looks at product characteristics. For example, a consumer may want a boat to be fast, but a designer may look at characteristics that affect the speed of a boat, such as engine size and hull shape. Kaul and Rao (1995) made the distinction between product *characteristics* and *attributes*. Product characteristics physically define the product and influence the formation of product attributes. Product attributes define consumer perceptions, and are usually fewer in number and more abstract than product characteristics. According to consumer decision theory, consumers compare products based on attributes, while designers look at product characteristics. Therefore, our model uses distinct sets of consumer attributes and designer characteristics.

Past product design research has focused on optimization with respect to *either* consumer attributes *or* designer characteristics. But both the consumer and the designer will benefit from reaching a *more balanced* product design. Designing products based solely on consumer attributes ignores the relationship between attributes and characteristics. Using only designer characteristics does not adequately account for the consumer in the design process. A better approach to product design would account for the relationship between the attributes and characteristics (Kaul and Rao, 1995).

Our model accounts for this relationship. When consumer attributes and designer characteristics are first defined, a relationship between them is also developed. This relationship takes a form similar to a "house of quality" matrix (Hauser and Clausing, 1988). In this matrix, one consumer attribute corresponds to one or more designer characteristics. The relationship matrix could be used to optimize a product design based on consumer attributes, and then translate the consumer attributes into the appropriate designer characteristics. But our model takes this a step further by optimizing on *both* the attributes and characteristics *simultaneously*. Once the attributes, characteristics, and the relationship between them are defined, conjoint analysis is used to collect data from the two parties. This data is then used to help determine an optimal product design based on the total share-of-choices considering both consumer and designer requirements.

3. A share-of-choices model for multisource product design

While the focus of this paper is on a pruning heuristic, we first review our model for product design that maximizes the share-of-choices for both consumers and designers. The model was first presented in Tarasewich and Nair (1999).

The inputs to the model are utility part worths data for designers and consumers and a relationship matrix that relates characteristic and attribute levels between the part worths matrices. The model contains k designer part worths matrices $D(k)$ for each designer characteristic $k = 1, \ldots, K$. Each designer characteristic k has $j_k = 1, \ldots, J_k$ levels. The rows of each matrix correspond to the $i = 1, \ldots, N_1$ designers. Entries in the matrix, d_{ijk}, are the part worths of individual i for level j of characteristic k. There is also a designer utility matrix of constants $D(0)$, with entries d_{i0} for each individual i. These constants are derived along with the part worths during conjoint analysis.

A product design preferred by designers, P_D, can be chosen by specifying a level j_k^* for each designer characteristic k. A total utility for each designer i, u_i, can be derived by summing over each characteristic at the specified level and adding this to the constant value:

$$u_i = d_{i0} + \sum_{k=1}^{K} d_{ij_k^*k}.$$

The model also contains $\hat{k}$ consumer part worths matrices $C(\hat{k})$ for each consumer attribute $\hat{k} = 1, \ldots, \hat{K}$. Each consumer attribute $\hat{k}$ has $\hat{j}_{\hat{k}} = 1, \ldots, \hat{J}_{\hat{k}}$ levels. Each row in the matrix corresponds to one of the $\hat{i} = 1, \ldots, N_2$ consumers. Entries in the matrix, $c_{\hat{i}\hat{j}\hat{k}}$, are the part worths of individual $\hat{i}$ for level $\hat{j}$ of attribute $\hat{k}$. There is also a consumer utility matrix of constants, $C(0)$, with entries $c_{\hat{i}0}$ for each individual $\hat{i}$.

A relationship matrix M is used to relate the set of designer characteristics and levels to the set of consumer attributes and levels. This is necessary because designers use a different set of definitions and levels to specify a product design than consumers. $m_{jk\hat{j}\hat{k}}$ gives a weighting value to the relationship between level j of designer characteristic k and level $\hat{j}$ of consumer attribute $\hat{k}$. The higher the value, the stronger the relationship is. More details on the concept of a relationship matrix can be found in Tarasewich and Nair (1999).

There is an equivalent consumer product design $P_{C|D}$ (read as the consumer design C given the designer design D) which is determined using the predefined relationship matrix M. The weights of M are summed for each level of each consumer attribute to determine the equivalent consumer product design $P_{C|D}$ (defined by a level $\hat{j}^*_{\hat{k}}$ for each attribute $\hat{k}$) for a given designer product design P_D. If there are ties, an arbitrary decision is made as to which consumer level is chosen.

A total product utility for each consumer $\hat{i}$, $v_{\hat{i}}$, can be derived for the equivalent product design $P_{C|D}$:

$$v_{\hat{i}} = c_{\hat{i}0} + \sum_{\hat{k}=1}^{\hat{K}} c_{\hat{i}\hat{j}^*_{\hat{k}}\hat{k}}.$$

Maximizing share-of-choices, or market share, is realistic in many situations, such as during the redesign of a product when there is a current product to compare with a new product design. A status-quo product design utility $\tilde{u}_i$ is calculated for each designer based on a currently offered product design. A similar value $\tilde{v}_{\hat{i}}$ is calculated for each consumer based on his or her status-quo product.

A designer prefers product design P_D over the status-quo if $u_i > \tilde{u}_i$. Suppose the total number of designers who prefer P_D to the status-quo design is n_1. Similarly, a consumer prefers product design $P_{C|D}$ to the status-quo if $v_{\hat{i}} > \tilde{v}_{\hat{i}}$. Let the total number of consumers who prefer $P_{C|D}$ to the status-quo design be n_2.

We define the objective as identifying a design P_D so as to maximize z_{SC}, the total fraction of share-of-choices (market shares), where

$$z_{SC} = \frac{n_1}{N_1} + \frac{n_2}{N_2}.$$

We divide n_1 by N_1 and n_2 by N_2 because the number of consumers is usually much greater than the number of designers, and the use of proportions factors this difference in sample sizes of the two groups. If the preferences of one group are more important than the other, the fractions can each be multiplied by weighting factors. The two fractions are added together because we want to optimize based on the total preferences of both groups. There may be other objective functions we could use, but this one seems simple and intuitive, and works well for us. Clearly, $0 \leqslant z_{SC} \leqslant 2$, with larger numbers indicating that more designers and consumers prefer the chosen product design over the status-quo.

3.1. A share-of-choices integer programming formulation

The following is an integer programming formulation of the share-of-choices multisource product design problem.

Let there be a set of $\Omega = \{1, 2, \ldots, K\}$ of K designer characteristics; each characteristic $k \in \Omega$ has J_K levels from the set $\Phi = \{1, 2, \ldots, J_K\}$. Let there be N_1 designers from the set $\Theta = \{1, 2, \ldots, N_1\}$. There is also a set of $\Omega' = \{1, 2, \ldots, \hat{K}\}$ of $\hat{K}$ consumer attributes, each attribute $\hat{k} \in \Omega'$ has $\hat{J}_{\hat{K}}$ levels from the set $\Phi' = \{1, 2, \ldots, \hat{J}_{\hat{K}}\}$. Let there be N_2 consumers from the set $\Theta' = \{1, 2, \ldots, N_2\}$. There is a utility value $\tilde{u}_i$ for designer i for the status-quo designer product design P_D and a utility value $\tilde{v}_{\hat{i}}$ for consumer $\hat{i}$ for the equivalent status-quo consumer design $P_{C|D}$. a_1 and a_2 are weights that can be set to values other than 1 if more emphasis needs to be placed on the share-of-choices coming from the designers or consumers (although then z_{SC} may no longer be between 0 and 2). $m_{jk\hat{j}\hat{k}}$ is a weighting value for the relationship between level j of designer characteristic k and level $\hat{j}$ of consumer attribute $\hat{k}$. Variable x_i equals 1 if designer i prefers a particular product design to the status-quo, and 0 otherwise. Variable $\hat{x}_{\hat{i}}$ equals 1 if consumer $\hat{i}$ prefers a particular design to the status-quo, and 0 otherwise. Variable x_{jk} equals 1 if level j is chosen for characteristic k for a particular designer product design, and 0 otherwise. Variable $\hat{x}_{\hat{j}\hat{k}}$ equals 1 if level $\hat{j}$ is chosen for attribute $\hat{k}$ for a particular consumer design and 0 otherwise. The formulation is:

maximize

$$z_{\mathrm{SC}} = a_1\left(\frac{n_1}{N_1}\right) + a_2\left(\frac{n_2}{N_2}\right) \tag{1}$$

subject to

$$n_1 = \sum_i x_i, \tag{2a}$$

$$n_2 = \sum_{\hat{i}} \hat{x}_{\hat{i}}, \tag{2b}$$

$$\sum_j x_{jk} = 1, \quad k \in \Omega, \tag{3a}$$

$$\sum_{\hat{j}} \hat{x}_{\hat{j}\hat{k}} = 1, \quad \hat{k} \in \Omega', \tag{3b}$$

$$u_i = d_{i0} + \sum_k \sum_j d_{ijk} x_{jk}, \quad i \in \Theta, \tag{4a}$$

$$v_{\hat{i}} = c_{\hat{i}0} + \sum_{\hat{k}} \sum_{\hat{j}} c_{\hat{i}\hat{j}\hat{k}} \hat{x}_{\hat{j}\hat{k}}, \quad \hat{i} \in \Theta', \tag{4b}$$

$$u_i + (1 - x_i)Q > \tilde{u}_i, \quad i \in \Theta, \tag{5a}$$

$$v_{\hat{i}} + (1 - \hat{x}_{\hat{i}})Q > \tilde{v}_{\hat{i}}, \quad \hat{i} \in \Theta', \tag{5b}$$

$$\sum_{\hat{j}} \left(\hat{x}_{\hat{j}\hat{k}} \sum_k \sum_j m_{jk\hat{j}\hat{k}} x_{jk} \right) \geqslant \sum_k \sum_j m_{jk\hat{j}\hat{k}} x_{jk}, \quad \hat{k} \in \Omega', \tag{6}$$

$$x_i, \hat{x}_{\hat{i}}, x_{jk}, \hat{x}_{\hat{j}\hat{k}} \in \{0, 1\}.$$

Here the objective function (1) maximizes the sum of the fraction of consumers and designers (weighted by a_1 and a_2, if necessary) who prefer a particular product design over the status-quo design. Q is a very large number. Constraints (2a) and (2b) simply sum the designers and consumers who prefer a particular product over the status-quo; (3a) and (3b) ensure that only one level is picked from each characteristic or attribute in the profile of each designer product and the equivalent consumer product; (4a) and (4b) calculate the total utility for a designer or consumer product design; (5a) and (5b) ensure that a designer and

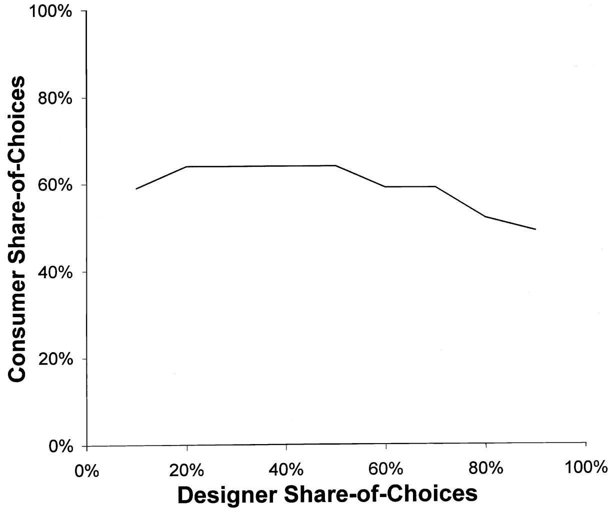

Fig. 1. Efficient frontier.

consumer prefer a design over the status-quo only if its utility is higher than the status-quo; (6) relates the set of designer characteristics and levels to the set of consumer attributes and levels. This constraint is quadratic.

An alternative approach can be used if a *set* of solutions (instead of a single point) is desired that shows the maximum consumer share-of-choices for each of designer share-of-choices value. This approach is appropriate if the decision-maker wishes to see the tradeoffs that can occur by placing more or less emphasis (weight) on the designers' preferences. In this situation, the IP formulation becomes:

maximize

$$z_{SC} = \frac{n_2}{N_2} \tag{7}$$

subject to

(2)–(6),

$$\frac{n_1}{N_1} = c. \tag{8}$$

If this formulation is used and c is varied between 0 and 1, we get a set of solutions (an "efficient frontier") such as the one illustrated in Fig. 1. The results in Fig. 1 are taken from a data set used in the simulation runs discussed in Section 5.

Next we describe an efficient way to find near-optimal solutions to our problem.

4. A pruning heuristic to design products using two perspectives

The number of possible product designs increases exponentially as the number of attributes and levels associated with a product increases. This makes the product design problem using conjoint analysis very difficult. In fact, the share-of-choices problem shown by Kohli and Krishnamurti (1989) to be NP-hard can be transformed in a polynomial number of steps to our share-of-choices problem with $N_1 = 0$. Hence our problem is also NP-hard. Heuristics to identify near-optimal product designs are a desirable alternative since complete enumeration to obtain optimal designs quickly becomes numerically prohibitive.

This section describes a solution method to the multisource data product design problem, something we call a "pruning heuristic". The heuristic first evaluates each level of each designer characteristic, and prunes away those levels it deems least likely to contribute to a good solution. A solution which maximizes the share-of-choices is then found through complete enumeration of the remaining levels. This method uses both designer and consumer data in its evaluation process, as well as the relationship matrix. Test results have shown that this heuristic produces near-optimal results at significant computational savings over complete enumeration of all attribute levels. It also performs well in many instances compared to other heuristic procedures such as Tabu search.

The steps of the pruning heuristic are as follows:

1. For each level j of each designer characteristic k, determine p_{jk}, which is the number of designers that prefer level j to the level of the status-quo product design for designer characteristic k. This is different from n_1, which is the number of designers that prefer an *entire* product design to the status-quo design.

2. For each level j of each designer characteristic k, determine $\hat{p}_{\hat{j}\hat{k}}$, which is the number of consumers that prefer level $\hat{j}$ to the level of the status-quo product design for consumer attribute $\hat{k}$.

3. Create a modified weighted total count, tc_{jk}, for each level j of designer characteristic k, where

$$\mathrm{tc}_{jk} = \frac{p_{jk}}{N_1} + \frac{\sum_{\hat{j}} \sum_{\hat{k}} \hat{p}_{\hat{j}\hat{k}} m_{jk\hat{j}\hat{k}}}{\mathrm{cc}_{jk} N_2 \sum_{\hat{j}} \sum_{\hat{k}} m_{jk\hat{j}\hat{k}}}.$$

cc_{jk} is the number of nonzero elements in column jk.

tc_{jk} estimates the relative importance of each designer level by adding the designer preferences to those of consumers through the preference matrix. The first term is simply the percentage of designers who prefer level j to the status-quo level for characteristic k. The second term first multiplies the weights in column jk of matrix M by the number of consumers who prefer level $\hat{j}$ to the status-quo level for attribute $\hat{k}$. The sum of these is then divided by the sum of the column weights multiplied by the total number of consumers multiplied by the number of nonzero elements in the column.

For example, using the following sample matrix M:

Consumer		Designer characteristic/level								
Attribute	Level	1			2			3		
		1	2	3	1	2	3	1	2	3
1	1	0	0	10	5	10	10	0	0	0
	2	10	5	5	5	0	0	0	0	0
	3	0	10	5	0	5	10	0	0	0
	4	5	0	10	10	0	5	0	0	0
2	1	0	10	5	0	0	0	10	5	0
	2	5	0	0	0	0	0	0	10	10
	3	0	5	10	0	0	0	5	5	5
	4	10	5	10	0	0	0	10	0	0

Assume $N_1 = 10$, $p_{11} = 7$, $N_2 = 100$, and

$$\hat{p} = \begin{bmatrix} 100 & 20 \\ 40 & 50 \\ 65 & 90 \\ 70 & 30 \end{bmatrix}.$$

We can see that $\mathrm{cc}_{11} = 4$ and

$$\mathrm{tc}_{11} = \frac{7}{10} + \frac{100(0) + 40(10) + 65(0) + 70(5) + 20(0) + 50(5) + 90(0) + 30(10)}{4(100)(0 + 10 + 0 + 5 + 0 + 5 + 0 + 10)} = 0.81.$$

4. Find

$$\overline{\mathrm{tc}}_k = \frac{\sum_j \mathrm{tc}_{jk}}{j},$$

the average weighted value for each designer characteristic k.

5. All levels for a given characteristic k with tc_{jk} less than ($\overline{\mathrm{tc}}_k * \mathrm{MOD}$) are eliminated, where MOD is used as a modification factor to fine-tune heuristic performance. MOD is the same for all characteristics. This means that a different number of levels can be eliminated from each characteristic, depending on how "bad" the levels appear. This also means that the total number of levels eliminated from each problem will vary. MOD is used to increase the probability of a good solution, but at the expense of higher computational time (a smaller MOD means fewer levels are eliminated, hence more designs enumerated). MOD was varied in this research between 0.9 and 0.7.

6. Enumerate all product designs using the characteristic levels not eliminated. The evaluation criterion for the chosen final design is z_{SC} (the total fraction of the share-of-choices of both parties, which is less than or equal to 2).

See Fig. 2 for the psuedocode for the share-of-choices pruning heuristic.

4.1. Computational complexity

The share-of-choices pruning heuristic has a complexity of

$$\mathrm{O}\left(N_1 \tilde{J}^K\right), \quad \text{where } \tilde{J} = \left(1 - \frac{\mathrm{MOD}}{2}\right) J.$$

```
Begin
    Determine p_jk for each level j of characteristic k
    Determine p̂_ĵk̂ for each level ĵ of attribute k̂
    Determine cc_jk for each level j of characteristic k
    Determine tc_jk for each level j of characteristic k
    Determine tc̄_k for each characteristic k
    For each characteristic k, eliminate levels where tc_jk < tc̄_k * MOD
    Enumerate product designs using remaining levels of k
        characteristics
    Determine z_sc for each enumerated product design
End.
```

Fig. 2. Pseudocode for share-of-choices pruning heuristic.

The expression $(1 - \mathrm{MOD}/2)$ accounts for the fraction of J levels eliminated from each attribute K before the remaining levels are completely enumerated. When MOD equals 1, approximately half of all levels will be eliminated before enumeration. As MOD approaches 0, virtually no levels will be eliminated. As MOD approaches 2, nearly all the levels will be eliminated. The expression J^K is simply the total number of enumerations completed if all J levels of all K attributes are used. Multiplying by N_1 accounts for the number of designers used in the problem. Clearly, the pruning heuristic is not polynomial. But it is still superior to complete enumeration, which has a complexity of $O(N_1 J^K)$.

The next section presents results of our heuristic using test data.

5. Computational results

In this section, the performance of the pruning heuristic is compared against complete enumeration using simulated data for the share-of-choices problem.

Complete enumerations of product designs using the share-of-choices model and simulated data were performed. The experimental design parameters were set as follows:

- Designer characteristics (k) – 5, 6, 7, 8, 9.
- Designer levels (j) – 4, 5.
- Designers (N_1) – 5%, 10% (of number of consumers).
- Consumer attributes ($\hat{k}$) – 3, 4, 5, 6.
- Consumer levels ($\hat{j}$) – 3.
- Consumers (N_2) – 100, 200, 400.

Not all combinations of parameters were run. Each run was done with 10 different starting seed values. The sizes of the problems ranged from 1024 (5 characteristics, 4 levels) possible product designs to 1,953,125 (9 characteristics, 5 levels) possible designs. Uniformly distributed random part worths (component utilities) varying between −5 and 5 were generated. The set of part worths for each level of a given characteristic or attribute for each designer or consumer sums to zero. These utility values closely resemble those found in actual data sets. A different relationship matrix was used for each problem size, but the matrix stayed consistent across the changing number of designers and consumers. Status-quo designs were also randomly generated for each problem size, but stayed consistent across the number of designers and consumers. The maximum possible share-of-choices (z_{SC}) is 2.

A second set of runs was done using the same set of parameters, except with normally distributed random part worths varying between −5 and 5. This was done for comparison purposes to the uniform data because some of our real data (from applications of our model) seems to be normally distributed. The results for normal data seem to follow the same pattern and distribution as those for uniform data.

The primary purpose in generating these complete enumerations is to compare them to heuristic results, but some general observations can be made. Values are generally increasing as we increase the problem size. This may occur because the increasing number of characteristics, attributes, and levels available make higher-percentage solutions possible. It also may simply be caused by the different relationship matrices and status-quo designs being used with the different problem sizes. It is difficult to accurately compare the problem sizes (rows of the tables) because the matrices are all different. There is a decrease in the total percentages as the number of consumers and designers increases. There are also higher values when using a smaller percentage of designers. This may occur because when there are fewer designers in the problem, it is easier to satisfy a higher percentage of them.

Next the pruning heuristic was run with the same sets of data. We varied MOD from 0.9 to 0.7 in increments of 0.05. Results for uniform data are presented in Tables 1 and 2. The tables differ based on the designer to consumer ratio. The results are given as the average percentage of the optimal solution achieved.

Table 1
Pruning heuristic results (Average percentage of optimal solution (uniform data, 1:10 designer ratio))[a]

	MOD = 0.9			MOD = 0.85			MOD = 0.8			MOD = 0.75			MOD = 0.7		
Designers	10	20	40	10	20	40	10	20	40	10	20	40	10	20	40
Consumers	100	200	400	100	200	400	100	200	400	100	200	400	100	200	400
Size															
5/4, 4/3	0.83	0.80	0.62	0.95	0.97	0.83	0.97	0.98	0.99	0.97	1.00	1.00	1.00	1.00	1.00
5/5, 4/3	0.96	0.99	0.99	0.99	1.00	1.00	0.99	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00
6/4, 4/3	0.97	0.96	0.90	0.98	0.97	0.97	0.98	1.00	0.99	0.99	1.00	1.00	0.99	1.00	1.00
6/5, 4/3	0.97	0.97	0.93	0.99	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
7/4, 3/3	0.92	0.88	0.83	0.95	0.95	0.92	0.96	0.97	0.99	0.99	0.99	1.00	1.00	0.99	1.00
7/4, 4/3	0.94	0.92	0.89	0.95	0.95	0.96	0.97	0.98	0.99	1.00	0.99	1.00	1.00	0.99	1.00
7/5, 4/3	0.97	0.95	0.97	0.98	0.99	0.98	0.99	0.99	0.99	0.99	1.00	1.00	0.99	1.00	1.00
8/4, 4/3	0.93	0.94	0.93	0.97	0.96	0.96	0.97	0.99	0.98	0.99	0.99	1.00	1.00	1.00	1.00
8/5, 5/3	0.98	0.97	0.97	0.98	0.98	0.99	0.99	0.99	1.00	1.00	0.99	1.00	1.00	1.00	1.00
9/4, 3/3	0.94	0.89	0.76	0.97	0.96	0.96	1.00	0.98	0.99	1.00	0.98	1.00	1.00	0.99	1.00
9/4, 4/3	0.94	0.94	0.93	0.96	0.98	0.96	0.98	0.99	0.99	0.99	0.99	1.00	1.00	1.00	1.00
9/5, 6/3	0.98	0.98	0.97	0.98	0.99	0.98	0.99	1.00	0.99	1.00	1.00	1.00			
Average	0.94	0.93	0.89	0.97	0.97	0.96	0.98	0.99	0.99	0.99	1.00	1.00	1.00	1.00	1.00

[a] Size = # designer attributes/levels, # consumer attributes/levels. MOD = modification factor (levels are eliminated if they are below average ∗ MOD). All data given are averages of 10 replicates.

Table 2
Pruning heuristic results (Average percentage of optimal solution (uniform data, 1:20 designer ratio))[a]

	MOD = 0.9			MOD = 0.85			MOD = 0.8			MOD = 0.75			MOD = 0.7		
Designers	5	10	20	5	10	20	5	10	20	5	10	20	5	10	20
Consumers	100	200	400	100	200	400	100	200	400	100	200	400	100	200	400
Size															
5/4, 4/3	0.90	0.83	0.82	0.95	0.95	0.96	0.99	0.97	0.98	0.99	0.97	1.00	1.00	1.00	1.00
5/5, 4/3	0.98	0.96	0.98	1.00	0.99	0.99	1.00	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00
6/4, 4/3	0.98	0.98	0.95	0.98	0.98	0.97	0.98	0.99	0.99	0.99	1.00	1.00	0.99	1.00	1.00
6/5, 4/3	0.98	0.98	0.98	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
7/4, 3/3	0.90	0.92	0.88	0.93	0.95	0.96	0.97	0.97	0.98	0.98	0.99	0.99	0.99	1.00	1.00
7/4, 4/3	0.92	0.95	0.93	0.96	0.96	0.96	0.97	0.97	0.98	0.98	1.00	1.00	0.99	1.00	1.00
7/5, 4/3	0.96	0.97	0.96	0.98	0.98	0.99	0.99	0.99	0.99	0.99	0.99	1.00	0.99	0.99	1.00
8/4, 4/3	0.97	0.92	0.94	0.99	0.98	0.97	0.99	0.98	0.99	0.99	0.99	1.00	0.99	1.00	1.00
8/5, 5/3	0.99	0.98	0.98	1.00	0.99	0.99	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00
9/4, 3/3	0.94	0.94	0.89	0.99	0.97	0.96	0.99	0.99	0.98	1.00	1.00	0.99	1.00	1.00	0.99
9/4, 4/3	0.97	0.95	0.94	0.99	0.97	0.97	0.99	0.99	0.99	1.00	0.99	0.99	1.00	0.99	1.00
9/5, 6/3	0.99	0.99	0.99	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00			
Average	0.96	0.95	0.94	0.98	0.98	0.98	0.99	0.99	0.99	0.99	0.99	1.00	1.00	1.00	1.00

[a] Size = # designer attributes/levels, # consumer attributes/levels. MOD = modification factor (levels are eliminated if they are below average $*$ MOD). All data given are averages of 10 replicates.

The results of the pruning heuristic are encouraging. With MOD set to 0.8, results are within 96% of optimal for all the test data sets. Good results are also obtained with MOD set to 0.85, especially with the lower ratio of designers to consumers (Table 2). Note that very good results are obtained with all MOD values for problems with 5 designer levels. Results seem to improve as problem size increases. Results worsen with a greater number of consumers, but this effect lessens with increased MOD values. Larger problem sizes have more levels to work with, so chances are greater that more good levels will be left for enumeration purposes after pruning and will provide better results. Likewise, if a problem has more levels per characteristic, chances are greater that enough good levels will be left after pruning for a given characteristic to provide good enumeration results.

The pruning heuristic was also run for the sets of normal data. The results of the normal data are very similar to the uniform data and are not presented here.

The results of the pruning heuristic are now compared against complete enumeration, a Genetic Algorithm based heuristic (Tarasewich and Nair, 1999), and Tabu search. Genetic Algorithms (GA) use biological methods such as reproduction, crossover, and mutation to quickly search for solutions to complex problems (e.g. Holland, 1975). Tabu search is a metaheuristic procedure that employs flexible memory structures, tabu restrictions and aspirations, and memory functions of different time spans. (Glover, 1990).

Performance results for 100 consumers and 10 designers in a variety of parameter settings for the three search procedures are shown in Table 3. Results given are the average percentages of the optimal solutions achieved, except for the run times, which are average values in seconds. The run times shown in Table 3 are how long a heuristic run or complete enumeration takes to produce a solution (machine time, not CPU times). For each problem size, the number of iterations that the Tabu search heuristic performed was adjusted to make its run times equivalent to the GA-based heuristic (with $W = 200$) to allow easy performance comparison between the two methods and the pruning heuristic. Times are meant for comparison purposes only. A reasonable effort was made to code efficiently in each case, but there may be other ways to implement the heuristics.

Results of the pruning heuristic are encouraging. Over the 10 different sets of starting strings, results for the values of MOD in Table 3 range from 95% to about 100% of optimal. There does not seem to be any change in performance of the pruning heuristic as problem size increases. Results of the pruning heuristic also compare favorably to the GA-based heuristic and to Tabu search, especially as the problem sizes become larger.

With an MOD value of 0.8, which is shown to provide good results, run times for the pruning heuristic compare very favorably with complete enumeration. Run times for the GA-based heuristic and Tabu search become better than run times for the pruning heuristic as problem size increases, but their performance begins to deteriorate. Unlike the GA-based heuristic and Tabu search, the pruning heuristic provides good results *and* decreased run times for smaller problems.

As MOD decreases, as can be seen from Table 3, the performance of the pruning heuristic improves. This is a result of more levels being enumerated. But decreasing MOD dramatically increases computation time. For small problems, the pruning heuristic seems to be the better approach. For larger problems, the GA-based heuristic may be the best overall choice, at least with a large enough value for W. The pruning heuristic, however, seems to do better overall than Tabu search, even with higher values of MOD (e.g. $\mathrm{MOD} = 0.85$) which result in faster run times.

There are no consistent differences between the results of the data with the number of designers set at 5% of the number of consumers versus the data having the number set at 10%. The values for the normal data follow the same patterns as those for the uniform data. Sensitivity analysis was also performed on the Tabu search heuristic by varying the sample size and the tabu list length. The parameters used for the comparisons in Table 3 are the ones that performed best based on this analysis.

A comparison across increasing numbers of consumers and designers is presented in Table 4. Values seem to remain fairly constant as the number of consumers and designers increase. The pruning heuristic

Table 3
Performance comparison with GA-based, Tabu search, and enumeration (Average percentage of optimal solution (uniform data, 100 consumers, 10 designers))[a]

Size	Pruning heuristic			GA-based heuristic			Tabu search	Run times (s)		
	MOD = 0.85	MOD = 0.8	MOD = 0.75	$W = 50$ $R1 = 10$	$W = 100$ $R1 = 20$	$W = 200$ $R1 = 20$	$n = 4$	EN	PR	GA, TS
5/4, 4/3	0.95	0.97	0.97	0.97	0.99	1.00	0.96	14	3	26
5/5, 4/3	0.99	0.99	0.99	0.94	0.97	0.99	0.93	46	7	32
6/4, 4/3	0.98	0.98	0.99	0.95	0.98	0.99	0.96	60	8	32
6/5, 4,3	0.99	1.00	1.00	0.94	0.97	0.99	0.96	246	37	36
7/4, 3/3	0.95	0.96	0.99	0.94	0.96	0.97	0.96	202	7	31
7/4, 4/3	0.95	0.97	1.00	0.94	0.97	0.98	0.97	262	11	39
7/5, 5/3	0.98	0.99	0.99	0.93	0.97	0.98	0.96	1343	145	46
8/4, 4/3	0.97	0.97	0.99	0.93	0.97	0.97	0.96	1175	60	48
8/5, 5/3	0.98	0.99	1.00	0.92	0.94	0.97	0.95	8455	643	54
9/4, 3/3	0.97	1.00	1.00	0.92	0.96	0.98	0.94	3740	70	40
9/4, 4/3	0.96	0.98	0.99	0.90	0.95	0.97	0.94	4576	102	57
9/5, 6/3	0.98	0.99	1.00	0.91	0.94	0.97	0.94			
Average	0.97	0.98	0.99	0.93	0.96	0.98	0.95	1829	99	40

[a] Size = # designer attributes/levels, # consumer attributes/levels; MOD = modification factor; W = working matrix size (in rows); $R1 = R2 = R3$; PR = pruning heuristic (MOD = 0.80); EN = complete enumeration; GA = GA-based heuristic ($W = 200$); TS = Tabu search; Pruning heuristic and Tabu search runs consist of 10 replicates; n = sample size; GA-based runs consist of 40 replicates (4 different mutation rates); Tabu list size was 4, 5, and 6 for problems 1–6, 7–10, and 11–12, respectively.

Table 4
Comparison for 100 and 200 consumers (Average percentage of optimal solution (uniform data))[a]

Size	Pruning heuristic		Bi-directional heuristic		Tabu search	
	$N_2 = 100$	$N_2 = 200$	$N_2 = 100$ $R1 = 20$	$N_2 = 200$ $R1 = 20$	$N_2 = 100$ $n = 4$	$N_2 = 200$
5/4, 4/3	0.99	0.97	1.00	1.00	0.96	0.99
5/5, 4/3	1.00	0.99	0.99	0.99	0.93	0.96
6/4, 4/3	0.98	0.99	0.99	0.99	0.96	0.97
6/5, 4,3	1.00	1.00	0.99	0.98	0.96	0.96
7/4, 3/3	0.97	0.97	0.97	0.98	0.96	0.95
7/4, 4/3	0.97	0.97	0.98	0.98	0.97	0.96
7/5, 5/3	0.99	0.99	0.98	0.97	0.96	0.95
8/4, 4/3	0.99	0.98	0.97	0.98	0.96	0.97
8/5, 5/3	1.00	1.00	0.97	0.97	0.95	0.95
9/4, 3/3	0.99	0.99	0.98	0.97	0.94	0.94
9/4, 4/3	0.99	0.99	0.97	0.98	0.94	0.97
9/5, 6/3	1.00	1.00	0.97	0.97	0.94	0.94
Average	0.99	0.99	0.98	0.98	0.95	0.96

[a] Size = # designer attributes/levels, # consumer attributes/levels; $N_1 = 0.1 * N_2$; $W = 200$ rows; $R1 = R2 = R3$; MOD = modification factor = 0.8; N_2 = number of consumers; Pruning heuristic and Tabu search runs consists of 10 replicates; n = sample size; Bi-directional runs consist of 40 replicates (4 different mutation rates); Tabu list size was 4, 5, and 6 for problems 1–6, 7–10, and 11–12, respectively.

provides slightly better results overall than the GA-based heuristic, and still better results overall than Tabu search, and in reasonable completion times for smaller problems.

6. Structural results

We now present some mathematical proofs to explain certain behaviors of the pruning heuristic and the general product design problem

6.1. Relationship between MOD and share-of-choices for pruning heuristic

MOD is used with the share-of-choices pruning heuristic to modify the number of levels of the given attributes that are eliminated. Here it will be formally proven that as MOD increases, the share-of-choices for a given problem will remain the same or decrease.

Start with a part worths matrix $D(k)$. Create a part worths matrix $X(k) \subset D(k)$. $X(k)$ is a subset of columns, *not* rows. Define the share-of-choices for $D(k)$ as $S_{D(k)}$, and for $X(k)$ as $S_{X(k)}$.

Lemma 1. *If $X(k) \subset D(k)$, then $S_{X(k)} \leqslant S_{D(k)}$.*

Proof. We will prove the result by contradiction. Suppose $S_{X(k)} > S_{D(k)}$. Then $X(k)$ has a better solution than $D(k)$, which means that $X(k)$has a set of part worths different from $D(k)$ which allow for a better solution.

But if $X(k) \subset D(k)$, then $X(k)$ must contain only part worths originally found in $D(k)$. Hence, any product design in $X(k)$ is also a product design in $D(k)$. Thus, the best design in $X(k)$ is a design that can be made from $D(k)$. Since the rows (number of designers) of the two matrices are identical, $S_{X(k)}$ has to be at most $S_{D(k)}$, which is a contradiction. Hence, $S_{X(k)} \leqslant S_{D(k)}$. □

Define a matrix $Y(k)$ with a share-of-choices value $S_{Y(k)}$. Introduce a number $\mathrm{MOD}_{X(k)}$, such that $X(k)$ is determined by $\mathrm{MOD}_{X(k)}$ and $D(k)$, and $Y(k)$ is determined by $\mathrm{MOD}_{Y(k)}$ and $D(k)$. If $\mathrm{MOD}_{X(k)} > \mathrm{MOD}_{Y(k)}$, then $X(k) \subset Y(k)$.

Theorem 1. *If* $\mathrm{MOD}_{X(k)} > \mathrm{MOD}_{Y(k)}$, *then* $S_{X(k)} \leqslant S_{Y(k)}$.

Proof. Since $\mathrm{MOD}_{X(k)} > \mathrm{MOD}_{Y(k)}$, $X(k) \subset Y(k)$. Therefore, using Lemma 1, $S_{X(k)} \leqslant S_{Y(k)}$. □

This says that as we increase the value of MOD in our pruning heuristic, the share-of-choices will decrease or remain constant.

6.2. Relationship between number of levels and share-of-choices

Lemma 1 and Theorem 1 looked at the effect that eliminating levels from an attribute part worths matrix has on its share-of-choices. We now look at what happens to the share-of-choices if we *increase* the number of levels for a given attribute.

For example, suppose an attribute for color has the levels red and green, and suppose red is preferred over green. Now another level, yellow, is added to the attribute. Red is still preferred over green, but yellow may be preferred over one, both, or neither of the other two colors.

Red, green, or both may be preferred to the status-quo color. This determines the original share-of-choices for the attribute. When yellow is added as a choice, the share-of-choices will increase if yellow is preferred over the status-quo, or stay the same if it is not.

Define $Z(k)$, which contains the part worths for all the levels from $D(k)$ plus part worths for one or more additional levels. The relative preferences of one level over another in $Z(k)$ are the same as in $D(k)$. Define $S_{Z(k)}$ for $Z(k)$.

Lemma 2. *For all* $Z(k)$, $S_{Z(k)} \geqslant S_{D(k)}$.

Proof. We will prove the result by contradiction. Suppose $S_{Z(k)} < S_{D(k)}$. Then by adding at least one more level to $D(k)$, less designers now prefer any possible product design from $Z(k)$ to the status-quo.

But $Z(k)$ contains the same levels and part worths as $D(k)$. Hence, at least as many designers who preferred a design from $D(k)$ should prefer a product design from $Z(k)$. Therefore, at the very least, $S_{Z(k)} = S_{D(k)}$, which is a contradiction. Hence $S_{Z(k)} \geqslant S_{D(k)}$. □

This says that as the number of levels of a given attribute increases, the share-of-choices will remain the same or increase.

Something similar cannot be done for the number of attributes. The share-of-choices for an entire product is based on a person having a total product utility greater than the total utility of the status-quo. If an attribute is eliminated, it is not known how much (or in what direction) the total utility is affected.

7. Summary

This paper has reviewed the importance of multisource product design and has reviewed a model that can be used to maximize the share-of-choices for a product designed using the preferences of both consumers and designers. In order to efficiently obtain near-optimal solutions to this problem, we have developed a heuristic based on pruning techniques. The effectiveness of the heuristic has been demonstrated through the use of test data and comparison to complete enumeration, a GA-based heuristic, and Tabu search. In many cases, the pruning heuristic performs well when compared to other search methods, and is a definite improvement over complete enumeration. Structural results for the pruning heuristic have also been provided. We expect to continue further development and testing of the heuristic, and to apply the model and heuristic to real problems.

Acknowledgements

The authors would like to thank Suresh K. Nair for his comments on earlier versions of this manuscript.

References

Balakrishnan, P.V., Jacob, V.S., 1996. Genetic algorithms for product design. Management Science 42 (8), 1105–1117.
Glover, F., 1990. Tabu search: A tutorial. Interfaces 20 (4), 74–94.
Hauser, J.R., Clausing, D., 1988. The house of quality. Harvard Business Review 66 (3), 63–73.
Holland, J.H. 1975. Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI.
Kaul, A., Rao, V.R., 1995. Research for product positioning and design decisions: an integrative review. International Journal of Research in Marketing 12 (4), 293–320.
Kohli, R., Krishnamurti, R., 1987. A heuristic approach to product design. Management Science 33 (12), 1523–1533.
Kohli, R., Krishnamurti, R., 1989. Optimal product design using conjoint analysis: Computational complexity and algorithms. European Journal of Operational Research 40, 186–195.
Kohli, R., Sukumar, R., 1990. Heuristics for product-line design using conjoint analysis. Management Science 36 (12), 1464–1478.
Nair, S.K., Thakur, L.S., Wen, K., 1995. Near optimal solutions for product line design and selection: Beam search heuristics. Management Science 41 (5), 767–785.
Tarasewich, P., Nair, S.K., 1999. Multisource product design using genetic algorithm based heuristics. Working paper. Maine Business School, University of Maine, Orono, Maine 04469. IEEE Transactions on Engineering Management, under review.