Sawtooth Software

RESEARCH PAPER SERIES

Efficient Experimental Designs Using Computerized Searches

Warren F. Kuhfeld,
SAS Institute, Inc.
1997

530 W. Fir St.
Sequim, WA 98382
(360) 681-2300
www.sawtoothsoftware.com

EFFICIENT EXPERIMENTAL DESIGNS USING COMPUTERIZED SEARCHES

Warren F. Kuhfeld
SAS Institute, Inc.

Abstract

In the past few years, marketing researchers have been increasingly using sophisticated computerized search algorithms to find experimental designs. This paper reviews some fundamentals of experimental design, orthogonality, and balance, and introduces the idea of design efficiency. It then compares some widely available design software including Sawtooth Software's CVA and SAS Institute's OPTEX programs.

Introduction

Conjoint analysis is used to study product purchase decisions when the products have several attributes or factors. Consumers "consider jointly" all of the attributes of a set of products, make trade offs, and then report their preferences for the products. The design of experiments is a fundamental part of conjoint analysis. Experimental designs are used to construct the hypothetical products.

For much of the history of experimental design and statistics, researchers used orthogonal designs that they looked up in tables. When an ANOVA model is fit with an orthogonal design, the parameter estimates are uncorrelated, which means each estimate is independent of the other terms in the model. More importantly, orthogonality *usually* implies that the coefficients will have minimum variance and hence maximum precision. For these reasons, orthogonal designs are usually quite good. ANOVA was widely used before the widespread availability of modern computers. With orthogonal designs, relatively simple formulas were available for hand or calculator ANOVA computations. Even as late as the 1970's, this was an important reason to use orthogonal designs. However, in the last ten to twenty years, general linear models software that does not require orthogonality has become widely available, so orthogonality is not as important as it used to be.

Like ANOVA, the early history of conjoint analysis is based on orthogonal designs. However, for many practical problems, particularly in marketing research, orthogonal designs are simply not available. Examples:

- when there are many attributes
- when the number of attribute levels is different for most of the factors
- a nonstandard number of cards is desired
- when some combinations are unrealistic, such as of the best product features at the lowest price.

In these and other situations, *nonorthogonal* designs must be used. During the past several years, marketing researchers are increasingly using efficient nonorthogonal designs (Kuhfeld, Tobias, and Garratt, 1994). These designs are efficient in the sense that the precision of the parameter estimates is maximized. Efficient designs can be found with the aid of a computer for nonstandard situations in which there are no orthogonal designs. A computerized search, with software such as the Sawtooth Software CVA (Conjoint Value Analysis) designer or the SAS Institute (1995) OPTEX procedure can be used to find good, efficient, and realistic conjoint designs.

Before exploring experimental design in detail, it is instructive to compare forms of conjoint analysis. CVA can be used to perform standard full-profile conjoint analysis where subjects rank or rate one product at a time. CVA can also be used for pair-wise presentation of products where subjects are asked to compare two products. CVA is typically used for paper and pencil administered studies. ACA (Adaptive Conjoint Analysis) is another widely used method for conjoint analysis. ACA interactively administers a conjoint study, adapting its questions to the individual respondent. ACA was designed for problems that generally could not be handled by full-profile methods, such as larger problems. CBC (Choice Based Conjoint) is used for fitting a multinomial logit model to discrete choice data. CBC, like ACA, collects data interactively, directly administering the study on the computer. However, CBC also has a paper-and-pencil module. ACA adapts its questions to the respondent; CBC and CVA do not.

CVA creates an efficient conjoint experiment using a computerized search. ACA does not attempt to create an optimal design. Instead, it is guided by another criterion, asking maximally informative questions. For choice models, it is impossible to create an efficient design without first knowing the "true" parameters. Hence, the construction of choice designs must be guided by other principles. CBC strives to make sure that for each pair of attributes, each level is paired with each other level (at least nearly) equally often.

Orthogonal Experimental Designs

An experimental design is a plan for running an experiment. The *factors* of an experimental design are variables that have two or more fixed values, or *levels.* Experiments are performed to study the effects of the factor levels on the dependent variable. In a conjoint study, the factors are the attributes of the hypothetical products or services, and the response is preference or choice. For example, Price could be a factor with levels $1.49, $1.99, and $2.49. A design is *orthogonal* if all effects can be estimated independently of all of the other effects (excluding the intercept). A design is *balanced* when each level occurs equally often within each factor, which means the intercept is orthogonal to each effect. Imbalance is a generalized form of nonorthogonality, which increases the variances of the parameter estimates.

A full-factorial design consists of all possible combinations of the levels of the factors. For example, with five factors, two at two levels and three at three levels $(denoted\, 2^2 3^3)$ [1], there are 108 possible combinations. In a full-factorial design, all main effects, all two-way interactions, and all higher-order interactions are estimable and uncorrelated. A full-factorial design is balanced and orthogonal. The problem with a full-factorial design is that, for most practical

---

1

$2^2 3^3\, means: 2 - level\ factors^{(there\,are\,2\,of\ them)}\ 3 - level\ factors^{(there\,are\,3\,of\ them)}$

situations, it is too cost-prohibitive and tedious to have subjects rate all possible combinations. For this reason, researchers often use *fractional-factorial designs*, which have fewer cards than full-factorial designs. The price of having fewer cards is that some effects become confounded. Two effects are *confounded* or *aliased* when they are not distinguishable from each other.

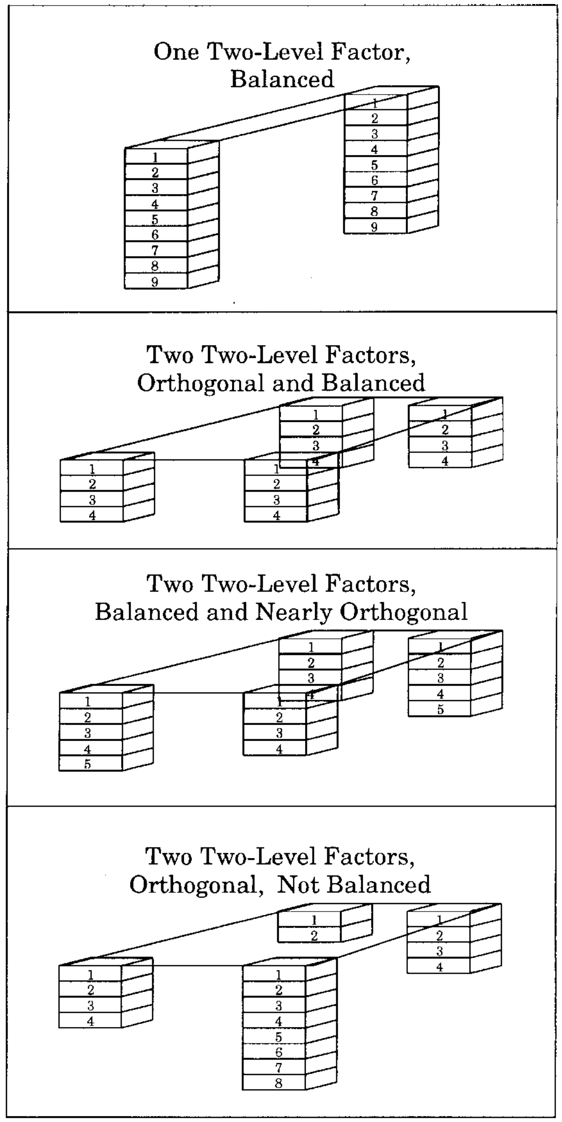
A special type of fractional-factorial design is the *orthogonal array,* in which all estimable effects are uncorrelated. Orthogonal arrays are categorized by their resolution. The resolution identifies which effects, possibly including interactions, are estimable. For resolution III designs, all main effects are estimable free of each other, but some of them are confounded with two-factor interactions. For resolution V designs, all main effects and two-factor interactions are estimable free of each other. Higher resolutions require larger designs. Orthogonal arrays come in specific numbers of cards (such as 16, 18, 20, 24, 27, 28, …) for specific numbers of factors with specific numbers of levels. Resolution III orthogonal arrays are frequently used in marketing research. The term "orthogonal array," as it is used in practice, is imprecise. It refers to designs that are both orthogonal and balanced, and hence optimal. It also refers to designs that are orthogonal but not balanced, and hence potentially nonoptimal.

Orthogonality and Balance

A good metaphor for discussing experimental designs is a raft. A raft is a flat boat that you hope will support your weight and keep you from getting wet. An experimental design forms the basis of a conjoint study, and you hope it will provide you with good information to support your marketing decisions. If your raft is not properly constructed, you will fall in the water and get eaten by alligators. If your experimental design is nonoptimal, you will have less information to use to make important decisions, and if your decisions are wrong, you will be eaten alive by your competitors.

A design with a single two-level factor is like a board, a one-dimensional raft, supported by Styrofoam floats. For maximum stability with N_D floats (cards), put $N_D/2$ floats under each end of the board. A design constructed according to this principle is balanced. If you put floats in the middle or more floats on one end, the board will be less stable. See figure 1.

Figure 1. Balance and Orthogonality, Illustrated with Rafts



Two two-level factors are like an ordinary square raft, supported by Styrofoam floats. For maximum stability, with N_D floats (cards), put $N_D / 4$ floats under each corner of the raft. A design constructed according to this principle is orthogonal and balanced. If you put floats in the middle or more floats on some corners, the raft will be less stable. Sometimes it is not possible to equally support all corners. Consider N_D 18 with two two-level factors. Then the best you can do is four cards with the (*a*, *a*) combination, four cards with the (*b*, *b)* combination, five cards with the (*a*, *b*) combination, and five cards with the (*b*, *a*) combination. A design constructed according to this principle is balanced and nearly orthogonal. Orthogonal designs can be very unbalanced. This leads to much less information being collected about some combinations than others. See Figure 1.

With three-level factors or more than two factors, the raft analogy is harder to imagine. However, the principles are the same. In orthogonal and balanced designs, the corners of the design space are well supported and equally supported. Nearly orthogonal and balanced designs where the corners are nearly equally supported are often the best that you can do in practice.

Coding

Before a design is used, it must be coded. One standard coding is the *binary* or *dummy variable* or (1, 0) coding. Another standard coding is *effects* or *deviations from means* or (1, 0, -1) coding. For evaluating design efficiency, we prefer an *orthogonal coding.* Standard nonorthogonal codings such as effects or binary coding are generally correlated, even for orthogonal designs. We use orthogonal codings so that we can get efficiency statistics scaled to range from 0 to 100. Efficiencies computed using nonorthogonal codings will have a smaller range (except for the special case of two-level factors).

- First a column of ones is coded for the intercept.

- A two-level factor *(a, b) is* replaced by one column.

Binary coding:	*a*	*is* replaced with	1
	b		0
Effects coding:	*a*	*is* replaced with	1
	b		-1
Orthogonal coding:	*a*	*is* replaced with	1
	b		-1

- A three-level factor (*a, b, c*) is replaced by two columns.

Binary coding:	*a*	is replaced with	1	0
	b		0	1
	c		0	0
Effects coding:	*a*	is replaced with	1	0
	b		0	1
	c		-1	-1
Orthogonal coding:	*a*	is replaced with	1.224745	-0.707107
	b		0	1.414214
	c		-1.224745	-0.707107

- A four-level *(a, b, c, d)* factor is replaced by three columns.

Binary coding:	*a*	is replaced with	1	0	0
	b		0	1	0
	c		0	0	1

	d		0	0	0
Effects coding:	*a*	is replaced with	1	0	0
	b		0	1	0
	c		0	0	1
	d		-1	-1	-1
Orthogonal coding:	*a*	is replaced with	1.414215	-0.816497	-0.57735
	b		0	1.632993	-0.57735
	c		0	0	1.73205
	d		-1.414214	-0.816497	-0.57735

- The orthogonal coding for an n-level factor is found by creating an *n x n* matrix C, with an intercept column and *n* - 1 columns containing the effects coding, then creating $\sqrt{n}C(C'C)^{-1/2}$ and discarding the first column.

Design Efficiency

Efficiencies are measures of design goodness. Common measures of the efficiency of and $(N_D\, x\, p)$ orthogonally coded design matrix **X** are based on the information matrix $X'X$. The variance-covariance matrix of the vector of parameter estimates β in a least-squares analysis is proportional to $(X'X)^{-1}$. The variance of $\hat{\beta}i$ is proportional to the x_{ii} element of .$(X'X)^{-1}$. An efficient design will have a "small" variance matrix, and the eigenvalues* of $(X'X)^{-1}$ provide measures of its "size." Two common efficiency measures are based on the idea of "average variance" or "average eigenvalue". *A-efficiency is* a function of the arithmetic mean of the variances, which is given by trace $((X'X)^{-1})/p$. (The trace is the sum of the diagonal elements of $(X'X)^{-1}$, which is the sum of the variances and is also the sum of the eigenvalues of $(X'X)^{-1}$.) *D-efficiency is* a function of the geometric mean of the eigenvalues, which is given by $\left|(X'X)^{-1}\right|^{1/p}$. (The determinant, $\left|X'X)^{-1}\right|$, is the product of the eigenvalues of $(X'X)^{-1}$, and the *pth* root of the determinant is the geometric mean.) A third common efficiency measure, *G-efficiency, is* based on σ_M the maximum standard error for prediction over the candidate set. All three of these criteria are convex functions of the eigenvalues of $(X'X)^{-1}$ and hence are usually highly correlated.

A-efficiency is based on the average of the variances of the parameter estimates. A-efficiency is perhaps the most natural criterion to use in evaluating design goodness. As orthogonality decreases, both the off-diagonal and diagonal elements of $(X'X)^{-1}$ increase. Looking at the average variance while ignoring the off-diagonal covariances, is reasonable because the variances increase as the covariances increase. D-efficiency is perhaps less intuitive

X * Eigenvalues are proportional to squared lengths. To understand eigenvalues, visualize a slightly deflated American football. Imagine holding it so the longest dimension is horizontal. Since it is partly deflated, imagine it positioned so the next longest dimension is vertical, and the shortest dimension corresponds to depth, the direction perpendicular to horizontal and vertical. The squared horizontal length is the first eigenvalue, the squared vertical length is the second eigenvalue, and the squared depth length is the third eigenvalue. These three numbers provide information about the size of the space occupied by the football. The eigenvalues of a variance matrix give information about the sizes of the variances.

than A-efficiency, but both provide a measure of the average size of the variance matrix. D-efficiency is used more often in practice for two reasons. Relative D-efficiency** is invariant under different codings; relative A-efficiency is not. Also D-efficiency is easier to update, so programs based on D-efficiency run faster.

For all three criteria, if a balanced and orthogonal design exists, then it has optimum efficiency; conversely, the more efficient a design is, the more it tends toward balance and orthogonality. Assuming an orthogonally coded **X**:

- A design is balanced and orthogonal when $(X'X)^{-1}$ is diagonal.
- A design is orthogonal when the submatrix of $(X'X)^{-1}$, excluding the row and column for the intercept, is diagonal; there may be off-diagonal nonzeros for the intercept.
- A design is balanced when all off-diagonal elements in the intercept row and column are zero.
- As efficiency increases, the absolute values of the diagonal elements get smaller and the diagonals approach $1/N_D$.

These measures of efficiency are scaled to range from 0 to 100:

$$\text{A-efficiency} = 100 \times \frac{1}{N_D \, trace((X'X)^{-1})/p}$$

$$\text{D-efficiency} = 100 \times \frac{1}{N_D \left|(X'X)^{-1}\right|^{1/p}}$$

$$\text{G-efficiency} = 100 \times \frac{\sqrt{p/N_D}}{\sigma_M}$$

These efficiencies measure the goodness of the design relative to hypothetical orthogonal designs that may be far from possible, so they are not useful as absolute measures of design efficiency. Instead, they should be used relatively, to compare one design to another for the same situation. Efficiencies that are not near 100 may be perfectly satisfactory.

---

8 ** Relative efficiency is the ratio of two efficiency statistics.

Figure 2. Candidate Set and Optimal Design

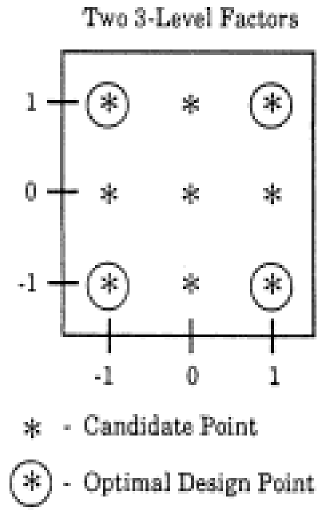


Figure 2 shows an optimal design in four cards for a simple example with two factors, using interval measure scales for both. There are three candidate levels for each factor. The full-factorial design is shown by the nine asterisks, with circles around the optimal four design points. As this example shows, efficiency tends to emphasize the corners of the design space. Interestingly, nine different sets of four points form orthogonal designs - every set of four that forms a rectangle or square. Only one of these orthogonal designs is optimal, the one in which the points are spread out as far as possible.

Computer-Generated Design Algorithms

When a suitable orthogonal design does not exist, computer-generated nonorthogonal designs can be used instead. Various algorithms exist for selecting a good set of *design points* from a set of *candidate points.* The candidate points consist of all of the factor level combinations that may potentially be included in the design, for example the nine points in Figure 2. For small problems, such as $2^2 3^3$, a good candidate set is the full-factorial design, since it contains only 108 cards. For larger problems, fractional-factorial designs make good candidate sets. When the full-factorial is more than say 1024 cards, it is always a good idea to try a fractional-factorial candidate set. Even with software that can handle several thousand candidates, it is good to also try small good candidate sets, because it is easier for the computer to find good designs when the search is limited to a small region.

N_D, the number of cards, is chosen by the researcher. Unlike orthogonal arrays, N_D can be any number as long as $N_D \geq p$, where *p* is the number of parameters.* The algorithm searches the candidate points for a set of N_D design points that is optimal in terms of a given efficiency criterion. There usually is not enough time to list all N_D-run designs and choose *the* most efficient or optimal design. For example, with $2^2 3^3$ in 18 cards, there are 108! / (18!(108 - 18)!) = 1.39 x 10^{20} possible designs. Instead, nonexhaustive search algorithms are used to generate a

* The number of parameters is the sum across all attributes of the number of levels of each attribute, minus the number of attributes, plus one for the intercept.

small number of designs, and the most efficient one is chosen. Usually, an initial design is randomly selected from the candidates, then it is iteratively refined. The algorithms select points from the candidate set for possible inclusion or deletion then update the efficiency criterion. The points that most increase efficiency are added to the design. These algorithms invariably find efficient designs, but they may fail to find *the* optimal design, even for the given criterion. For this reason, we prefer to use terms like *information-efficient* and *D-efficiency* over the more common *optimal* and *D-optimal*.

There are many algorithms for generating information-efficient designs. We will begin by describing some of the simpler approaches and then proceed to the more complicated (and more reliable) algorithms. Dykstra's (1971) sequential search method starts with an empty design and adds candidate points so that the chosen efficiency criterion is maximized at each step. This algorithm is fast, but it is not very reliable in finding a globally optimal design. Also, it always finds the same design (due to a lack of randomness). These next algorithms are typically run repeatedly for a given candidate set and different random initial designs, then the most efficient design is chosen. The Mitchell and Miller (1970) simple exchange algorithm is a slower but more reliable method. It improves an initial design by adding a candidate point and then deleting one of the design points, stopping when the chosen criterion ceases to improve. The DETMAX algorithm of Mitchell (1974) generalizes the simple exchange method. Instead of following each addition of a point by a deletion, the algorithm makes excursions in which the size of the design may vary. These three algorithms add and delete points one at a time.

The next two algorithms add and delete points simultaneously, and for this reason, are usually more reliable for finding the truly optimal design; but because each step involves a search over all possible pairs of candidate and design points, they generally run much more slowly (by an order of magnitude). The Federov (1972) algorithm simultaneously checks each candidate point and design point pair, then makes the swap that most increases efficiency. Cook and Nachtsheim (1980) define a modified Federov algorithm that checks each candidate point and design point pair and makes every swap that increases efficiency. The resulting procedure is generally as efficient as the simple Federov algorithm in finding the optimal design, but it is up to twice as fast.

CVA Designer

The CVA design software automatically: creates a candidate set, excludes prohibited pairs, creates an initial design, uses the modified Federov algorithm to improve the efficiency, then it discards the candidate set and performs additional iterations to improve balance and overall efficiency. It repeats this process a user-controlled number of times (five by default) then outputs the best design. Here is more detail on the algorithm:

- CVA generates the candidate set with a guided randomization process. For each attribute, CVA randomly picks a pair of levels from all permitted pairs that have been presented least often. Pairs of levels are not repeated until all other permitted pairs have been shown. This creates a candidate set with good balance. For example, when a 20-profile design is requested, CVA by default creates a candidate set with 120 profiles.

- Next, CVA creates an initial design. It starts with the full candidate set and excludes one card at a time, the card that contributes the least to the design. CVA considers excluding each point and checks its effect on efficiency. It performs the exclusion that leads to the maximum efficiency. At first, efficiency may actually increase as the points that provide the least information are removed. Then, typically, efficiency will start to decrease. The initial design has the same number of profiles as the final design—for example 20.

- Next, CVA uses the modified Federov algorithm, swapping (previously excluded) candidates back into the design until efficiency stops improving.

- For the final step, the candidate set is discarded. CVA looks for imbalance and identifies the levels that appear most often. CVA considers changing those levels, making sure that prohibited pairs are not introduced. If changing a level to improve balance also increases efficiency, it is done. In effect, CVA is using a virtual candidate set in this step. Possible candidates include every card in the full factorial (minus prohibited pairs), but only a relatively few candidates are considered, those that improve balance.

- The entire process is repeated and the best design is chosen.

We will investigate the CVA designer for use in full-profile conjoint experiments. Other capabilities of CVA such as its ability to generate designs for pair-wise presentation are beyond the scope of this paper.

The OPTEX Procedure

The OPTEX procedure requires the user to create a candidate set. A good candidate set for a small problem is a full-factorial design. Resolution III, IV, V, and perhaps larger designs are good candidate sets for larger problems (Kuhfeld, 1996). Unrealistic or undesirable combinations can then be excluded from the candidates. PROC OPTEX starts with a random initial design and then iteratively improves it. PROC OPTEX has sequential, exchange, DETMAX, and Federov algorithms, but I usually use Modified Federov. The entire process is repeated (10 times by default), and the best design is chosen.

An Empirical Evaluation of CVA and PROC OPTEX

This section compares CVA and PROC OPTEX with problems. The first three examples are plausible conjoint studies. The next two examples are harder problems than you are likely to find in a reasonable conjoint study.

The first test was a relatively simple problem, $2^2 3^3$ in 18 cards. The optimal design, described by Kuhfeld, Tobias, and Garratt (1994), is nonorthogonal. CVA requires the user to enter the names of the factors, the levels, and the number of cards. CVA generates the candidate set and performs the searches. I created a full-factorial design for the PROC OPTEX candidate set. Both CVA and PROC OPTEX found the optimal design, with D-efficiency = 99.861 and perfect balance, in a matter of seconds.

The next test was harder, $2^2 3^3 5^2$ in 30 cards, but still small enough to be realistic for a conjoint experiment. CVA found a good design with D-efficiency = 96.2149 in about three minutes. The balance was perfect. All of my attempts with CVA to find a better design, by both generating more designs and changing the size of the candidate set failed. Using PROC OPTEX, I was able to find an unbalanced design with D-efficiency = 97.6690. With subsequent tries, I found a perfectly balanced design with D-efficiency = 98.0327. Since the full-factorial design at 2700 cards is large, I started with fractional-factorial candidate sets and worked my way up to the full factorial. The CVA design was slightly (98%) less efficient than the PROC OPTEX design, and both programs found perfectly balanced designs.

The next test was harder still, $2^2 3^3 5^2 7$ again in 30 cards. CVA found a good design with D-efficiency = 88.2956, which I was able to increase to 89.8463 in subsequent tries. The balance was excellent. With PROC OPTEX, I found designs with efficiency ranging up to 92.2954. The CVA design was slightly less efficient than the PROC OPTEX design but much better balanced.

Next, I tried a larger and much more difficult problem, 2 3 4 5 6 7 8 9 10 11, using the CVA recommended 168 cards. This is not a realistic design for a full-profile conjoint analysis (at least without blocking). Still, it seemed reasonable to test CVA's performance with a larger and more difficult problem. CVA found a design with efficiency 94.3472, whereas PROC OPTEX found a design with efficiency 96.3529. Again, the PROC OPTEX design was slightly more efficient, and the balance in the CVA design was excellent and much better than the PROC OPTEX design.

Last, I tried a large problem, $2\,3\,4^2 5^2 6$ with 24 cards* and prohibited pairs. CVA allows the user to specify pairs of attribute levels that should never be presented together, for example largest size and smallest price. PROC OPTEX allows any combination to be excluded from the candidate set. The following pairs were prohibited: (x l = 1, x2 = 1), (x2 = 1, x3 = 1), (x3 = 1, x4 = l),(x4 = 1, x5 = l), (x5 = 1, x6 = 1), and (x6 = 1, x7 = 1). Using a full-factorial candidate set and generating ten designs, PROC OPTEX found a design with D-efficiency = 86.9362 in eight minutes. Generating 100 designs took one hour and resulted in a D-efficiency of 88.4463. Balance was good but not perfect. The first level tended to occur less often, particularly in the two- and three-level factors due to the prohibited pairs. I easily found a CVA design with D-efficiency = 81.9513. Letting CVA run overnight resulted in D-efficiency = 84.2510. The CVA design was better balanced than the OPTEX design.

CVA is easier to use than PROC OPTEX, particularly for the less-experienced user, because CVA automatically creates the candidate set. In contrast, for the more-experienced user, PROC OPTEX is more likely to find a more efficient design because the user can control the candidate set. PROC OPTEX typically runs faster than CVA, but with more user set-up time. For all but very small problems, PROC OPTEX is typically run several times with different candidate sets, then the best design from the best candidate set is chosen. For difficult problems, there are many ways to create reasonable candidate sets, and it is impossible to predict which way will work best. Learning how to create good candidate sets is not easy.

CVA usually finds good designs with excellent balance. For small problems like you would typically encounter in a full-profile conjoint study, CVA seems to do an excellent job. However, for larger and more difficult problems, it often fails to find more efficient designs that can be found with PROC OPTEX. The differences in efficiency between the two programs are small and may in part be offset by CVA's better balance. Balance is very important. You do not want any level (particularly for attributes like brand and price) appearing a lot more often than some other level. Some analysts generate many designs, output the most efficient few, and them pick

* This design is almost saturated since there are 23 parameters, so this example is not realistic.

the most balanced design from the most efficient few designs, even if the most balanced design is not the most efficient. In the next section, I will discuss ways in which CVA and PROC OPTEX might be improved.

An Evaluation of CVA and PROC OPTEX Algorithms

CVA starts by creating a guided random candidate set with good balance. It then creates an initial design by excluding cards from the candidate set. The approach typically used with PROC OPTEX is for the user to create a full-factorial design for small problems, and resolution III, IV, V, and perhaps larger candidate sets for larger problems. PROC OPTEX by default uses a random sample of the candidate set as the initial design. The next step for both methods is the modified Federov swaps, which is quite standard and works quite well. CVA has a final step that iterates further, simultaneously increasing efficiency and improving balance. This last CVA step is new, innovative, and I believe a *very* good idea.

The reason that PROC OPTEX can often find a more efficient design than CVA is due to the first two steps. I suggest that Sawtooth Software seriously look at using a full factorial for the candidate set with small problems and fractional factorials for larger problems. Candidate sets with well over one thousand cards should not pose any problems on today's PC, although it is frequently the case that a smaller candidate set is actually better. Perhaps using CVA's guided randomization to augment a core fractional-factorial candidate set would be a good idea. (I have never actually tried this.) I also suggest that Sawtooth Software consider using a random sample from the candidate set as the initial design.

For small problems, the CVA modified Federov swaps are reasonably fast. For larger problems I think they could be made faster. The final efficiency and balance optimization is no doubt the reason why CVA does such a good job of finding (at least nearly) balanced designs. However, it is slow for large problems and could be made faster.

PROC OPTEX would benefit from a graphical user interface, an option for automatic candidate set creation, and an option to optimize balance like CVA does.

Conclusions

Computer-generated experimental designs can provide both better and more general designs for conjoint studies. Classical designs, obtained from books or computerized tables, are good when they exist, but they are not the only option. When the design is nonstandard and when there are restrictions, a computer can generate a design, and it can be done *quickly*. For most conjoint studies, a good design can be generated in a few minutes. Furthermore, when the circumstances of the project change, a new design can again be quickly generated. The computerized search usually does a good job, it is easy to use, and it can create a design faster than manual methods, especially for the nonexpert. In nonstandard situations, simultaneous balance and orthogonality may be unobtainable. Often, the best that can be hoped for is optimal efficiency. Computerized algorithms help by searching for the most efficient designs from a potentially very large set of possible designs.

I am very pleased that more marketing researchers and more software packages are now using efficiency to guide their design search. PROC OPTEX does an excellent job in finding efficient designs even for very large problems, however less-experienced users may find it hard to use. CVA does an excellent job with small problems (which are the kinds of problems for which it was designed), a good job with larger problems, produces designs with excellent balance, and is particularly well suited for less experienced users. The final stage of the CVA designer algorithm is innovative, and does an excellent job of producing at least a nearly balanced design.

References

Cook, R. Dennis and Christopher J. Nachtsheim (1980), "A Comparison of Algorithms for Constructing Exact D-optimal Designs," *Technometrics,* 22 (August), 315-24.

Dykstra, Otto (1971), "The Augmentation of Experimental Data to Maximize $\left|(X'X)^{-1}\right|$," *Technometrics,* 13 (August), 682-88.

Federov, Valery V. (1972), *Theory of Optimal Experiments*, translated and edited by W.J. Studden and E.M. Klimko, New York: Academic Press.

Kuhfeld, Warren F., Tobias, Randall D., & Garratt, Mark (1994) "Efficient Experimental Design with Marketing Research Applications," *Journal of Marketing Research*, 31 (November), 545-557.

Kuhfeld, Warren F. (1996) "Marketing Research Methods in the SAS System," an unpublished collection of papers and handouts.

Mitchell, Toby J. and F.L. Miller, Jr. (1970), "Use of Design Repair to Construct Designs for Special Linear Models," *Math. Div. Ann. Progr. Rept.* (ORNL-4661), 130-31, Oak Ridge, TN: Oak Ridge National Laboratory.

Mitchell, Toby J. (1974), "An Algorithm for the Construction of D-optimal Experimental Designs," *Technometrics, 16* (May), 203-210.

SAS Institute, Inc. (1995), *SAS/QC Software: Reference, Version 6, First Edition*, Cary, NC: SAS Institute, Inc.