Sawtooth Software

TECHNICAL PAPER SERIES

Advanced Simulation Module (ASM) for Product Optimization Technical Paper

3210 N. Canyon Rd., Provo, Utah
+1 801 477 4700
www.sawtoothsoftware.com

Advanced Simulation Module (ASM)
for Product Optimization
Technical Paper & Software Examples

Introduction

Conjoint/choice analysis (i.e. CBC) is used by many organizations to assess the likely success of competing choice alternatives or potential new products/services. Conjoint analysis assumes that an individual's liking for an alternative can be approximated as the sum of part-worths for its separate attribute levels. A conjoint questionnaire involves a designed experiment for each individual, providing data on the tradeoffs among attribute levels from which it is possible to estimate his or her part-worths.

Those estimated part-worths can be used in choice (market) simulations. The simplest simulation specifies several competitive alternatives in terms of their attribute levels and then predicts which of those each respondent would prefer. By rolling up those individual choices across a population, we can estimate potential relative choice share for hypothetical new or modified products. If the conjoint experiment involves a pricing attribute, we can also estimate potential relative revenue and profitability.

Conjoint analysis has been enormously successful in many different fields and organizations, largely because of its choice simulation capability, since its introduction in marketing research more than 45 years ago (Green and Rao, 1971). Choice simulators convert part-worth data (the scaling of which can be difficult to understand for many decision-makers) into shares of choice summing to 100%—resembling market shares—that are easy to understand and immensely practical for managers.

Simulators represent the best conjoint/choice has to offer in terms of assessing attribute importance and sensitivities, complex interaction or substitution effects, and the likely success of products (services, or other choice alternatives) given certain competitive conditions.

Software System and Subscriptions

The ASM (Advanced Search Module) capabilities described in this technical paper are limited specifically to Sawtooth Software's Choice Simulator, which is a Windows-based software system. The Choice Simulator may be installed and launched in two different ways:

- As an integrated component within Lighthouse Studio when you have a conjoint analysis subscription with the ASM add-on permissions. After you have used the Analysis Manager to estimate part-worth utilities for your conjoint analysis project, you can click the Choice Simulator icon to launch the Choice Simulator.
- By you or by your client as a standalone Choice Simulator installation on a Windows machine. Once you're licensed for the ASM, you can install the standalone Choice Simulator by visiting http://www.sawtoothsoftware.com/support/downloads. Your clients do not need to purchase a subscription to the standalone Choice Simulator. If you

send them the choice simulator file (project.sim) where you as a licensed user have set up some search scenarios, then your client can use the Choice Simulator to open that .sim file and use/modify those scenarios.

Toward Optimization

The typical choice simulation involves a set of competing alternatives and asks, "How good would ***this*** alternative be?" Decision-makers and researchers often turn the question around: rather than ask "How good would ***this*** alternative be?" they often ask "What alternative would be ***best***?" The Advanced Simulator Module (ASM) has been developed to answer such questions.

Although conjoint/choice simulators can be used in a variety of fields and for studying many different kinds of choices or purchases, for the remainder of this document we'll refer to the common marketing case involving buying products.

Sawtooth Software is by no means the first to consider how one might search for products that optimize market share or profitability. Green and Krieger (1993) considered the same problem years ago, proposing a method similar in many ways to what we have done.

Choice simulators provide the best means to date for product optimization. They can take into account the characteristics of currently-available alternatives as well as the desires of a heterogeneous population of potential buyers or choosers. Subject to reasonable caveats about the quality of respondent sampling and questionnaire design, choice simulations can accurately assess likely product success long before a product is ready for test market. The ASM can optimize based on utility, purchase likelihood, market share, revenue or profitability. Profitability optimization requires additional user-provided information about feature costs. If you include feature costs, the ASM can also perform cost minimization searches, to search for products that meet some threshold of utility, share, revenue, or profit while minimizing cost.

Finding the best product by manually specifying many simulations would be difficult because there can be such a large number of potential products to evaluate. Suppose products are described by 10 attributes, each with 6 levels. Then the number of possible products (ignoring the possibility of interpolation between levels) would be 6 to the 10th power, over 60 million. Although each of those possible products could be simulated, examining all combinations would take too long to be workable. When we consider that one may wish to optimize multiple products simultaneously, even situations with few attributes would be infeasible for manual search.

Another possible approach would be to configure a product with attribute levels that are most desirable on average. But that would fail in two ways. First, it would almost certainly choose an unprofitable product because it would select products with many desirable features and the lowest possible price. The second reason such a method would fail is that it does not recognize heterogeneity in the desires of the market. Moreover, marketers know that the most successful products are those that appeal to buyers who are not already satisfied by existing products, and hence existing products must be taken into account.

The ASM uses either exhaustive or heuristic search strategies to find the optimum (or near-optimum) product or a portfolio (product line) of multiple products. It can optimize several kinds of objectives, including estimates of market share, total revenue, profitability, purchase likelihood, and total utility. It does this by exploring the response surface of the objective, such as share, corresponding to attribute levels for the product(s) of interest.

Although most response surfaces are multi-dimensional, it is useful to imagine there are only two independent variables. Suppose a product category has two attributes, both continuous, and that we are interested in market share. Then we could represent share estimates from many simulations by a three-dimensional model. The product attributes would be represented by the X and Y axes and the shares from simulations of each combination of attributes would be represented by the third dimension, the height of a surface. Our task is to find the highest point on that surface.

One strategy would be to imagine a grid underlying the surface and to measure its height at every grid point. This is a reasonable approach if there are few dimensions, but quickly becomes unworkable as the number of dimensions increases and the number of grid points multiplies. If the surface has a single peak, another strategy would be to pick a point on the surface, find the direction in which the surface rises most rapidly, and move in that direction to find a higher point, repeating the process until the highest point is found. Such steepest ascent hill-climbing methods are efficient with surfaces having single peaks, but can be misled if there are multiple peaks. Because different optimization strategies are more effective with different kinds of response surfaces, the ASM provides several from which to choose.

The ASM can be used to analyze conjoint data from any of Sawtooth Software's conjoint systems (CBC, ACBC, ACA or CVA). You can also use it to analyze conjoint analysis data from a 3rd party or your own customized approach as long as you format the part-worth utility data correctly for file import. The ASM can be used for full-profile, partial-profile, and alternative-specific designs. Estimation can include linear terms and interaction terms. The ASM can also be used with aggregate conjoint data, though it is most effective with individual data.

The paragraphs above have provided a general introduction without details; but several topics require closer examination. These include the types of product simulations available, the specific optimization methods provided, and the ways cost information is used in the calculation of profitability.

Simulation Methods

The Advanced Simulation Module provides several methods for simulating preferences. Here we shall provide just a brief description of each, ranging from simplest and fastest to slowest but most useful.

First Choice or Maximum Utility: This is the simplest simulation method. Each respondent's utility for each product is estimated by summing the appropriate part-worths. The utilities for all products are compared, and the respondent is assumed to choose the product with maximum utility. Another way to say this is that we assume all of a respondent's choice likelihood accrues to his *first choice* product, regardless of the magnitude of difference in utility between that product and the others. The estimate of a product's share of market is simply the percentage of respondents for whom it has highest utility.

This method has some desirable characteristics. It is simple and easy to understand. Also it is fast, so optimizations done with First Choice simulations proceed quickly. Most important, First Choice simulations are not vulnerable to difficulties caused by the inclusion of similar products, such as when evaluating portfolios of similarly branded alternatives. We'll describe this problem shortly.

However, first choice simulations also have some undesirable properties. They tend to exaggerate the shares of popular products and underestimate the shares of unpopular products. Further, unlike other methods, there is no way to tune them (such as via the *exponent*) to compensate for this characteristic. A second shortcoming is that since all of a respondent's choice likelihood is allocated to a single product, the standard errors of the resulting shares are larger than with other methods that distribute a respondent's choice likelihood across several products.

The First Choice method is mainly of historical interest, and we would not advocate using it in optimizations unless there are special circumstances, such as if simulating using multiple vectors of utilities (e.g. HB draws) per respondent, a compelling need for computational speed, the benefit of exceptionally large sample sizes, and confirmation that the relative scaling of share results is appropriate.

Share of Preference: This method is only slightly more complicated than the First Choice method. As with that method, each respondent's utility is computed for each product. However, rather than assigning all of a respondent's choice likelihood to the product with maximum utility, we allocate choice likelihood among products by first exponentiating all products' utilities (converting them to positive numbers) and then percentaging the results so that they sum to 100. (An example appears below.) This is equivalent to employing a logit model for product choice.

The Share of Preference method is nearly as fast as the First Choice method and has the additional benefit that the results can be tuned so the ratio of maximum to minimum estimated product shares can be adjusted as desired (e.g. the response error or scale factor in conjoint questionnaires may not match the choice errors in actual purchases). This is accomplished by multiplying the part-worths by a positive constant (the *exponent*). A large constant causes more extreme share predictions and in the limit the Share of Preference method approaches the First Choice method. A small constant causes shares to be more nearly equal, and with a very small constant the predicted shares will all become nearly equal.

However, as with all logit models, the Share of Preference method is vulnerable to what are known as *IIA problems*. (IIA is short for *Independence from Irrelevant Alternatives*). A simple example can demonstrate this. Suppose there are two quite different products (A and B) for which a respondent has utilities of 0 and 1.0. Then Share of Preference estimates of the respondent's choice likelihoods would be as below:

Product	Utility	Exponentiated Utility	Share Estimate
A	0.0	1.00	26.9%
B	1.0	2.72	73.1%

		3.72	

Now suppose we introduce another product, A', which has characteristics identical to those of product A. Then in a three-way simulation we obtain the following estimates:

Product	Utility	Exponentiated Utility	Share Estimate
A	0.0	1.00	21.2%
A'	0.0	1.00	21.2%
B	1.0	2.72	57.6%

		4.72	

We have nearly doubled the total estimated share of the A products merely by including a second copy. We could drive the estimated total share of the A products as high as we like simply by including many of them. But we know this is not a realistic simulation of real world conditions, where a respondent who prefers traveling via car to a bus is likely to choose the car no matter how many buses are available.

By contrast, the First Choice method assigns the respondent to product B no matter how many copies of inferior product A are included, demonstrating much more reasonable behavior.

With logit models, a newly introduced product takes share from existing products in proportion to their current shares. (The ratio of original shares 26.9 / 73.1 = 0.368 is the same as the ratio of new shares 21.2 / 57.6 = 0.368.) This property is useful in some circumstances, but presents problems in our context because a newly introduced product will probably *not* take share from others proportionally to their shares. For example, an additional package size for a soft drink may be expected to take more business from other sizes of its own brand than from other brands. We refer to this important property as differential substitution.

This IIA property of logit models is troubling, because we may wish to estimate the total market share for a portfolio of products that are somewhat similar to one another. We may wish to know how many more units of a make of car will be purchased if we offer a convertible in addition to a sedan, or how many more ounces of cereal will be purchased if we offer a jumbo package in addition to a regular sized one. In the case of aggregate models, such as a main effects logit model, the Share of Preference method is obviously inappropriate for questions like these.

The Share of Preference method should provide good results when all of the products in a simulation are equally similar to one another. However this is a hard condition to ensure, so it is more prudent to use a method that is not vulnerable to this difficulty.

IIA problems are substantially reduced when dealing with individual (e.g. utilities from HB) rather than aggregate simulations. Individual-level utilities usually have much larger magnitudes than aggregate utilities, so individual simulations often allocate nearly all of a respondent's preference to a single product such they become more like First Choice simulations. Thus, Share of Preference simulations are likely to be less misleading when used at the individual level. Still, as the number of product alternatives in a simulation scenario grows, the IIA difficulties become even more problematic.

The advanced setting for Share of Preference for distributing share only to the Top-N within each person is a simple way to potentially reduce IIA problems. For simulations involving many product concepts (such as 10 or more), the share for each respondents could be allocated to only (say) the top 3 product alternatives—whereas the remaining less preferred products could all receive 0 share.

Randomized First Choice (RFC): This is the default method for simulations involving choices among three or more competing products. It is slower than other methods, but nearly resolves the previously described shortcomings.

Results for each respondent are simulated many times in sampling iterations. The part-worths are perturbed randomly for each iteration. The perturbations are of two types. First, the part-worths themselves are perturbed (by adding *attribute error*) and the modified part-worths are used to sum utilities for each product. Then the utility sums themselves optionally may be further perturbed (by adding *product error*). For each iteration the respondent is allocated to the product with highest (perturbed) utility per the *First Choice* rule. The results for all iterations are averaged to produce the final estimate of choice share for each respondent.

It is clear that real respondents are somewhat inconsistent when making choices. This suggests that the values they ascribe to product features actually vary from moment to moment, and the RFC procedure is an attempt to mimic that variability.

This method is much slower than the preceding ones, because each respondent's choices are simulated many times rather than just once. However the method has compensating advantages.

The default for the RFC method is to add only *attribute* rather than *product* error. In that case IIA problems are avoided and similar products (in terms of sharing some or many attribute levels) do not receive inflated shares. This means that RFC simulations are appropriate for a wide range of occasions, including estimation of the value of line extensions and portfolios of similar products.

Also, unlike simple First Choice simulations, Randomized First Choice simulations can be tuned to provide differing ratios of extremity between shares for popular and unpopular products. The magnitudes of perturbations of part-worths and utilities can be adjusted. In general, as magnitude of perturbations is increased, predicted shares become more similar. In the limit, with very large perturbations and many sampling iterations, all products' shares would become equal. It is commonly found that First Choice simulation results are too extreme. One reason is that in the real world not every product is always available and occasionally a buyer must accept a product that would not otherwise be his or her first choice. Like Share of Preference simulations, and unlike First Choice simulations, Randomized First Choice simulations can be tuned (either by adjusting the magnitude of perturbations or using the *exponent*) to more closely mimic actual market shares.

Purchase Likelihood Simulations: Sometimes a product category is so new that there are no competitive products to which it may be compared. At such times it is useful to model purchase likelihood rather than share of market. Some conjoint methods which ask respondents to rate product alternatives on a purchase intent rating scale produce part-worths that are scaled so that utility sums can easily be converted to estimates of purchase likelihood. In that case a simulation could involve a single product, and if there are multiple products, their results are unaffected by one another (for example, all products could have high likelihoods, or low likelihoods).

Purchase Likelihood simulations can be very fast. But we caution the user not to interpret the results literally. Respondents are notoriously unable to estimate likelihoods of any kind, and likelihoods of purchase of new products are no exception. Purchase Likelihood estimates should never be interpreted as more than directional indicators of buyer preference. It is difficult for respondents to report reliability their absolute purchase intent.

Recommendation: We recommend Randomized First Choice as the preferred method in most circumstances. Exceptions include situations in which there are fewer than three product alternatives being simulated, and when only brand and price are involved. If a much faster method is required, the First Choice method has the advantage of not inflating shares for similar products, but its estimated market shares will probably be too extreme and their standard errors will be somewhat larger.

The Share of Preference method is also fast, and can be tuned to provide the desired amount of variation in product shares, but may inflate the shares of products that are similar to others, especially with a great many product alternatives in the scenario. This difficulty can be partially ameliorated by using Share of Preference with a Top-N setting (to assign non-zero shares of preference only to the top N product alternatives per the share of preference method).

If the product of interest is unique or too new for there to be a competitive product category, then Purchase Likelihood simulations may be appropriate, but only to provide a relative ranking of products rather than to provide accurate estimates of actual purchase likelihoods.

All simulation methods can be run either with respondents weighted equally or with respondents weighted by a variable selected by the user. Furthermore, external effects can be included, awareness may be defined per product by respondent, and the Exponent (scale factor) may be tuned, when appropriate.

Optimization Methods

To conduct product searches in the ASM, the user specifies several items:

- One or more products for which optimal attribute levels are to be discovered.
- Competitive products that will be held constant throughout the analysis (unless searches are to maximize utility or purchase likelihood for a single offering).
- The objective to be optimized (estimated market share, purchase likelihood, profitability, revenue, cost minimization, or total utility) whether for a single product or across a portfolio of products.
- For each attribute, the range of levels permitted and optionally “interpolation steps” or specific user-specified values for any attributes that are to be treated as continuous rather than discrete variables. The user may also specify ranges within attributes that will not be explored.
- If profitability searches or cost minimization are used, product cost information is also required. Attribute-based price information can also be included, so that the addition of certain product features is always accompanied by a set increase (or decrease) in price.

Three optimization algorithms are available, which differ in several ways. We now describe each of them briefly.

Exhaustive Search is the simplest of the algorithms. It examines every combination of permitted levels of all attributes. For example, if an attribute has three levels (levels 1, 2, 3) and interpolation for half-steps is used, there would be 5 levels of that attribute to explore (1, 1.5, 2, 2.5, 3).

The main strength of Exhaustive Search is that it is guaranteed to find the best solution (from among the domain specified). If the response surface has multiple peaks, Exhaustive Search will evaluate all of them and choose the best one.

The main shortcoming of Exhaustive Search is that the number of combinations to be evaluated can become very large. For example, with 10 attributes, each having 5 possible levels, the number of solutions to explore would be 5 to the 10^{th} power, or nearly 10 million.

Thus Exhaustive Search will probably not be used until faster methods have first reduced the size of the problem. For example, other methods might quickly determine that interest should be focused on narrower ranges of several attributes. If a particular level appears always to be present in the winning product, then you can hold that level constant, letting other attributes vary. After holding several attributes constant, Exhaustive Search could be used to find the best combination of levels within a reduced domain of possibilities.

Grid Search is much faster than Exhaustive Search. The attribute levels to be explored are specified in exactly the same way, but rather than examining all potential combinations, Grid Search proceeds heuristically. A starting solution is chosen at random. Then several iterations are conducted. Within each iteration the attributes are selected in random order, all permitted levels of each attribute are examined (with all other attributes held constant) and the best level is retained. In each iteration the attributes are examined in a different random order. The process continues until an iteration fails to find a better solution. That is to say, Grid Search stops after a solution has been found that cannot be improved by changing any single attribute.

The main strength of Grid Search is its speed. The number of combinations to be evaluated increases only linearly with the number of attributes and levels, rather than as a function of the numbers of levels raised to the power of the number of attributes. If the response surface is single-peaked, Grid Search is guaranteed to find the optimum. If there are multiple peaks, then repeated runs from different starting points are very likely to find it. Its main shortcoming is that Grid Search is not guaranteed to find the global optimum if there are several peaks. However, in our experience Grid Search is very effective at quickly finding good solutions, and thus can be used to reduce the domain required for further exploration by Exhaustive Search.

Genetic Search is our implementation of the procedure described by Balakrishnan and Jacob (1996). We start with a population of simulation solutions of size 300, consisting of potential solutions obtained by running individual-level steepest ascent grid optimizations on a randomly-selected group of 300 respondents (though random selection of simulation solutions is also a possible option in the software). In each iteration ("generation") the least "fit" half of the population is replaced with new members obtained by "mating" of the most fit 150 members. We assume that the most fit "parents" are most likely to produce fit "children." Parents for each new member are chosen randomly, with probability proportional to their fitness.

Each child has a combination of the parents' characteristics. For interpolable variables the child's value is a random one, rectangularly distributed between the parents' values. Also, the child's value is subjected to "mutation" by adding a normal random variable. For non-interpolable variables, the child receives the value of one parent or the other with equal probability.

As a measure of success we use the objective value for the best member of that generation. It is difficult to say when one should stop iterating. Lack of improvement in one generation does not necessarily mean that there will be no improvement in future generations. For single-objective optimization, the default is to stop when three successive generations have failed to show improvement, although the user may modify that setting—and we recommend doing so if it is important to find a potentially better result. Multi-objective genetic search is possible and is described in the documentation and in an article by Orme (2018).

Genetic Search has various strengths:

- Unlike Grid Search, it makes no assumptions about the shape of the response surface and is therefore not hindered by discontinuities due to constraints.
- Another is that it is theoretically possible that the final population may include near-optimal members who are high on different peaks, and in these cases Genetic Search is less vulnerable to local optima. In our initial work, we have seen this only occasionally, and for these unusual cases Genetic Search consistently returns good solutions. For the most part we find that the response surface is more generally unimodal and the faster "steepest assent" methods also consistently achieve near-optimal solutions.
- It can find a large number of near-optimal solutions for the researcher and the decision-maker to consider.
- It can be used in multi-objective searches, where the goal is to simultaneously consider more than one objective (e.g. share of preference and revenue) and find a result that offers an efficient tradeoff among multiple objectives (Orme 2018).

Genetic Search's main shortcoming is slowness. We have found it to take several times as long as the Grid method.

Recommendations: As a robust general-purpose approach, the Grid Search is very fast and obtains either the optimal or near-optimal solutions. To obtain even more confidence about a solution, it makes sense to run Genetic Search and examine the variation in the near-optimal attribute levels among the top 50 or more solutions. The experience obtained could permit reducing the size of the unknown domain substantially (to those levels represented in the top, say, 50 solutions), so it may become feasible to run Exhaustive Search in that reduced domain.

Genetic Search is a useful algorithm that has the potential of finding good solutions when conditions limit the capabilities of other methods, such as when the response surface is very irregular with multiple peaks. It takes much longer, but in certain cases achieves superior results.

For both Genetic and Exhaustive Search the user can specify the number of solutions that should be reported, and that many of the best solutions are displayed in the report window. This capability seems most likely to be useful when the response surface has multiple peaks. In that case the researcher may be able to use expert opinion to evaluate a number of near-optimal solutions.

We have seen situations where searches seem to capitalize on "reversals" in part-worths to produce less useful solutions. It may, therefore, make sense to use part-worths that have been constrained during estimation to avoid reversals.

Introducing Product Costs and Estimating Profitability

Profit is the universal measure of success for most businesses and as such is usually the most valuable search criterion. Unless constrained to search within relatively profitable spaces of opportunity, Utility, Share of Preference, or Purchase Likelihood simulations mostly produce trivial solutions where the best features are delivered at the cheapest prices (attribute-based pricing, described below, offers an exception). Moreover, revenue searches focus solely on revenue without regard to profit. We recognize that a firm may have a specific goal in mind for a particular product line, such as maximizing penetration. But these strategies are generally the exception rather than the rule.

If profitability is to be optimized, the user must provide information about the unit cost for features. Such information is often hard to obtain, but we urge users to avail themselves of cost data or to approximate costs wherever possible.

Cost information may be specified for attributes independently, or may vary depending on other variables. For example, if we were studying a pharmaceutical product, one attribute might be type of container, and another might be size of container. The costs of different types of container can depend on their sizes, and on several other variables as well, if desired. It is also possible to use advanced scripting to change the costs for attribute levels depending on the share of preference that the searched product achieves (e.g. to reflect economies of scale once production hits a certain level).

The arithmetic of the profitability computation is very simple. For each product we have not only part-worths reflecting respondent values, but we also have part-costs indicating the contribution to cost of different product attributes. We sum those part-costs to get the cost of one unit of product. We subtract that cost from the price of the product to get a unit margin and we multiply unit margin by estimated market share to get a measure of relative profitability. The user may specify the total number of units sold in the market, in which case estimated revenue and profitability can be stated in actual monetary amounts.

Attribute-Based Prices to Improve Optimization Searches

As mentioned previously, profit optimization searches are the most useful to a business. However, most consultants and firms probably will not have access to accurate cost data. Without cost data, searches often yield trivial solutions wherein the “optimal” product is the one with the best features at the lowest price. Solutions can be constrained to consider attribute combinations that are most realistic and promising, but this often doesn’t adequately address the problem.

Even though many firms do not have access to detailed costing information by attribute level, they often know (or can approximate) the incremental prices that different attribute levels add to the base price of an offering. Automobiles, computers, custom-built homes, and manufacturing equipment (just to mention a few) have different options (e.g. add $1,200 for air conditioning, add 1TB more hard drive space for $30), and buyers consider the utility of the options and their incremental prices in choosing a final product to purchase.

In the absence of good cost information, it makes sense to incorporate attribute-based pricing information, assuming good data are available. Thus, a desirable product feature is not automatically included within a searched product unless its marginal value exceeds the marginal price for buyers.

Caveats

One problem in the early years of conjoint analysis was that simulation results looked so much like actual market shares that managers sometimes forgot they were only estimates. The same difficulty applies to product optimizations.

Users must remember that optimization results are only the results of statistical exercises. There are numerous ways things can go wrong. To name just a few, (1) the sample of survey respondents must be chosen appropriately, be of sufficient size, and be relatively free from non-response bias, (2) the conjoint questionnaire must be designed well (include the right attributes and levels), (3) the competitive context must be specified accurately, and (4) the most robust methods should be used to estimate part-worths.

Results concerning revenue and profitability are especially troublesome. They depend critically on proper estimation of price coefficients. We know from years of experience that many conjoint methods do not do a good job of estimating price sensitivity. Of the methods provided by Sawtooth Software, CBC and ACBC are the most effective and ACA the least effective for pricing studies. However, as effective as CBC or ACBC can be in the hands of a knowledgeable researcher, it is easy to describe attribute levels in vague and unclear ways that lead to misleading results.

Finally, profitability computations require not only buyer preference information, but also cost information. That information is hard to come by, and simply may be unavailable in many organizations. Needless to say, the accuracy of profitability estimation depends critically on the accuracy of cost data. It also follows that the quality of the results when using attribute-based pricing also depends upon the price inputs.

As a final note, it is important to recognize that even if the caveats above are dealt with appropriately, the results of an optimization are only valid inasmuch as the assumed fixed competition does not react. When competitors react, the previous solution is usually no longer optimal.

How Well Does It Work?

We have tested the optimization routines with several conjoint data sets volunteered by Sawtooth Software users. We have been able to assume cost information, so our tests have involved all types of optimization criteria supported by the ASM.

The results with all data sets and search criteria have been similar, so we shall describe only one set of results in detail. This data set consists of conjoint part-worths for 546 respondents on 15 attributes having a total of 83 levels. We shall disguise the product category.

With this data set we constructed six hypothetical competitive products and sought characteristics for an optimal seventh product. Realizing that the highest-share product would necessarily have the lowest price, we constrained the price of the new product to a fixed level suggested as reasonable by the provider of the data set. One other attribute was also constrained, which dealt with the "form factor" of the product, since the owner of the data set already had a clear idea of what form factor he wished to explore.

We have subjected this data set to hundreds of optimizer runs, both to explore the comparative effectiveness of the several optimization procedures, and to "tune" default settings for several user-specifiable details.

We assess the effectiveness of each simulation method by two measures: (1) how reliably it obtains what we believe to be the best answer, and (2) how much computational effort is required. Since we have done our testing on computers with varying clock speeds, we standardize our time measurements by reporting the number of actual product simulations required.

Even after constraining two of the attributes to constant levels, there remain approximately 143 million possible product configurations. Even with the fastest of the simulation methods and on the fastest desktop computer available at the time (in 2003, when these simulations were conducted) the Exhaustive Search method would require nearly five days of continuous computation, clearly not an efficient alternative.

By contrast, to see how quickly a solution could be obtained, we ran Grid Search with Share of Preference simulations, limiting the domain explored to whole-number levels (using interpolation steps of 1). That computation required only 97 simulations, completed less than a second using a computer with a 2.8 GHz processor. The solution differed in only one attribute from the best solution found in hundreds of optimization runs using RFC simulations.

All further results we shall report use RFC simulations. We report results for seven runs of each optimization method, all from different random starting points.

We judged five attributes to be interpolable; that is to say, we explored the entire region between their maximum and minimum levels, not limiting our exploration to whole-numbered levels. Eight other attributes were judged not to be interpolable, and we explored only whole-numbered levels. Two attributes, price and form factor, were locked at fixed levels.

The Grid search algorithm got the same answer each time, which is also the best answer we have found by any method. Interestingly, it consisted of only whole-numbered levels, even for attributes treated as continuous. The median number of evaluations required by the Grid method was 187. Using a computer with 2.8 GHz processor, these runs required an average of a little less than a minute.

Genetic Search never got exactly that same answer, although it came close several times. The worst of its seven results had a product share 70% as high as the apparent optimum, and two of the seven runs had product shares 98% as high as the apparent optimum. Letting it run longer rather than halting after three generations without improvement might have yielded better answers, but at the cost of longer run times. As it was, the median number of simulations required by Genetic Search was 1600, implying run times about ten times as long as the Grid approach.

As a final check on the optimality of the apparently best solution, we ran Exhaustive Search on a restricted set of attribute levels, including all levels that had appeared in any of the solutions described so far, plus all whole-numbered levels between those. That run required 3600 simulations and about a 20 minutes on the same computer. The best solution it found was identical to that produced by the other methods.

Thus we can be quite confident that the best solution found was indeed optimal, although absolute certainty would require an impossibly long Exhaustive Search. The encouraging result is that this solution was also found seven times out of seven by Grid Search.

A problem with 15 attributes and 83 levels may be smaller than some that users of the software will encounter, but the computational time for those three algorithms tends to increase only linearly with the number of total attribute levels. Therefore problems with much larger numbers of attributes and levels (or including optimization of multiple products simultaneously) should still be solvable in reasonable time.

We should note that the findings from this one data set will not apply generally to all data sets. We have seen other data sets for which genetic search consistently provides near-optimal answers, and the “steepest ascent” techniques sometimes return noticeably sub-standard answers. Thus, we recommend you alternate between different methods, first starting with a quicker methods/settings and then working toward longer run times as you gain familiarity with the data set and/or reduce the search space.

Specifying Dynamic (Searched) Products

When using the market simulator in typical simulation mode, you specify multiple products using a spreadsheet-like grid. For example, you might specify three products as following:

Label	Brand	Screen Size	Inputs	Resolution	Picture in Picture	Price
Product 1	Spectre	37" screen	RCA (composite) + 1 HDMI	720p	No picture in picture	$400
Product 2	Vizio	37" screen	RCA (composite) + 2 HDMI	720p	Picture in picture	$500
Product 3	Vizio	42" screen	RCA (composite) + 2 HDMI	1080p	Picture in picture	$700

However, when searching for optimal products, you enter *ranges* of levels rather than fixed levels for the product to be searched. For example, in the grid below we specify that product 4 (“Searched 1”) must be Sony (our brand), but all the other values can vary.

Label	Brand	Screen Size	Inputs	Resolution	Picture in Picture	Price
Product 1	Spectre	37" screen	RCA (composite) + 1 HDMI	720p	No picture in picture	$400
Product 2	Vizio	37" screen	RCA (composite) + 2 HDMI	720p	Picture in picture	$500
Product 3	Vizio	42" screen	RCA (composite) + 2 HDMI	1080p	Picture in picture	$700
Searched 1	Sony	1-3	=Range(1,3)	1,2	=Range(1,2)	=range(400,700,50)

We’ve used the ***Range*** instruction in the example above for the searched product, along with some other examples of valid syntax. Here are two examples to illustrate the syntax for ***Range***:

- **=Range(100,300)** *indicates that search should examine all level values (levels included in the attribute level list) from 100 to 300*
- **=Range(100,300,10)** *indicates that search should examine all values from 100 to 300 using a step size of 10 units (100, 110, 120, etc. through 300)*

These examples are also valid ways to specify searched ranges for products:

- **1-3** *indicates that search should examine levels 1, 2, and 3 for this attribute*
- **1-3,5** *indicates that search should examine levels 1, 2, 3, and 5 for this attribute*
- **100, 125, 145.75, 175, 200, 240, 280, 300** *indicates that search should examine levels of 100, 125, 145.75, etc. through 300 (for a continuous attribute, where you have associated values of 100 to 300 with the levels included in your attribute list)*

You can search for multiple products simultaneously, such as finding the best 2 products to offer. In the example below, the product search will seek to maximize the *sum* of the shares, revenues, profits, or utilities of "Searched 1" and "Searched 2" products.

Label	Brand	Screen Size	Inputs	Resolution	Picture in Picture	Price
Product 1	Spectre	37" screen	RCA (composite) + 1 HDMI	720p	No picture in picture	$400
Product 2	Vizio	37" screen	RCA (composite) + 2 HDMI	720p	Picture in picture	$500
Product 3	Vizio	42" screen	RCA (composite) + 2 HDMI	1080p	Picture in picture	$700
Searched 1	Sony	37-40	=Range(1,3)	1,2	=Range(1,2)	=range(400,550,50)
Searched 2	Sony	40-42	=Range(1,3)	1,2	=Range(1,2)	=range(550,700,50)

Each of these two searched products is constrained to be brand level 1 (our brand). "Searched 1" is constrained generally to have smaller screen sizes and lower prices, whereas "Searched 2" is constrained to have larger screen size and higher prices. In the case of a multiple product line search problem, although it is not necessary to constrain searched products to have mostly unique searchable ranges, it often leads to quicker convergence on better results.

When conducting product searches, you can also specify that certain static (fixed) products should be included in the objective function (the value to be maximized, whether share, revenue, profit or utility). For example, you may have a current product on the market that will not change, but you want to search for another product to offer, such that the profit for the existing product combined with the new product are maximized.

Prohibiting Combinations of Levels

In the previous example, we showed how to constrain searched products to have certain characteristics (and to prohibit other characteristics). You will probably find that most prohibitions you wish to enforce can be done in the same manner as we've shown, using only the product entry grid with specific ranges to search specified for different products. If you need to specify additional prohibited combinations, you can do so within the software.

Entering Cost and Price Information

To perform profitability searches, cost minimization searches (exhaustive search only) or other searches (e.g. share, utility) subject to the criterion that cost cannot exceed some threshold, you are required to specify cost information. In the ASM, costs are associated with specific levels in your study design. It is not necessary to specify costs for all attributes in your study, or for all levels within a particular attribute. Some attributes may not have associated costs, and some levels may be irrelevant to your particular search (i.e. your competitor's costs). Costs can be tied to the levels of a single attribute, or on the joint combination of multiple attributes.

Often, cost information is not available, but pricing information is. The ASM lets you specify incremental prices associated with different attribute levels in your study. Including attribute-based prices for attributes reduces the likelihood that share, utility, or revenue optimizing solutions simply reflect the best features at the cheapest prices. A feature is only included if the marginal benefit is greater than the marginal price of the feature. The "part prices" assigned to attribute levels can be either positive or negative, and are added to the overall base price of searched products.

Practical Optimization Examples

Introduction

In this section, we'll illustrate some common optimization problems that can be solved using the ASM. In the first example, we specify step-by-step how to use the software to perform the searches. Following the first example, we'll describe additional types of problems more generally, to give you a greater understanding of the flexibility within the program.

The examples covered include:

1. New Product Introduction without Competition (Utility, Purchase Likelihood Search)
2. New Product Introduction with Competition (Share, Revenue Search)
3. Searching for Multiple Products Simultaneously
4. New Product Introduction with Competition (Profitability Search)
5. Searches Using Attribute-Based Prices
6. Maximize Appeal Subject to a Maximum Cost
7. Minimize Cost Subject to Meeting a Performance Threshold
8. Multi-Objective Search: Maximize Both Share of Preference and Profit

Example #1: New Product Introduction without Competition (Utility, Purchase Likelihood Search)

We'll begin with a simple illustration that doesn't involve a "market simulation" at all, but focuses on finding a single product configuration to maximize average utility for the population. This, admittedly, is a problem that doesn't require an optimization routine. But for the purpose of illustration, we'll begin with the simple case.

Imagine you are searching for an optimal vacation package. We'll refer to the "SampleACA" ACA (Adaptive Conjoint Analysis) project that ships with Lighthouse Studio and is used in the ACA documentation.

The average part-worths (using the OLS utilities, with the zero-centered Diffs rescaling method) for the 100 respondents in this data set are:

Label	Utility
Paris, France	11.42
New York City, USA	-24.75
Rome, Italy	20.97
London, UK	8.35
Sidney, Australia	11.84
Singapore	-27.83

4 nights	-33.49
6 nights	9.31
8 nights	24.18
2-star hotel	-72.38
4-star hotel	20.38
5-star hotel	52.00
Business-type hotel	-32.71
Resort-type hotel	32.71
Coach accommodations on airplane (economy)	-32.79
Business class accommodations on airplane (extra leg room)	0.84
First Class accommodations on airplane (premium space & services)	31.95
Most direct flight possible	43.53
1 extra 3-hour layover (beyond most direct route)	-3.23
1 extra 6-hour layover (beyond most direct route)	-40.29

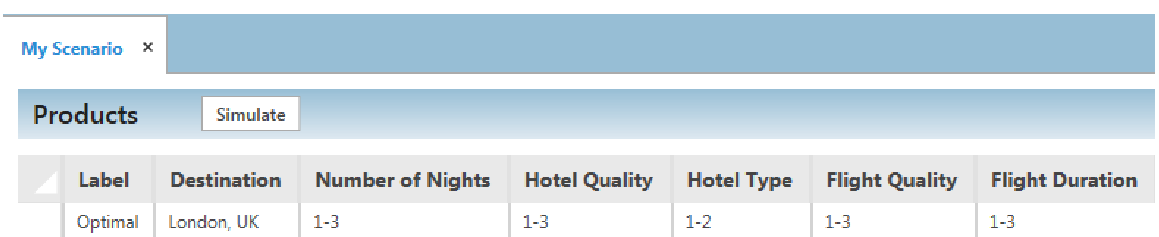
With this array of attributes and levels, there are 6x3x3x2x3x3= 972 possible product combinations. Let's assume the destination is fixed: London. With destination fixed, there are now 3x3x2x3x3= 162 remaining combinations.

If the goal is to provide the single, most appealing London vacation (assuming no competition) without regard to cost, the optimal product is trivial. After destination (Rome, Italy), one selects the most preferred level on average from each remaining attribute:

Level	**Utility**
London, UK	8.35
8 nights	24.18
5-star hotel	52.00
Resort-type hotel	32.71
First class accommodations	31.95
Most direct flight possible	43.53

Total:	192.71

Even though the answer is trivial, we can use the ASM to search for this result. We use the following specifications:

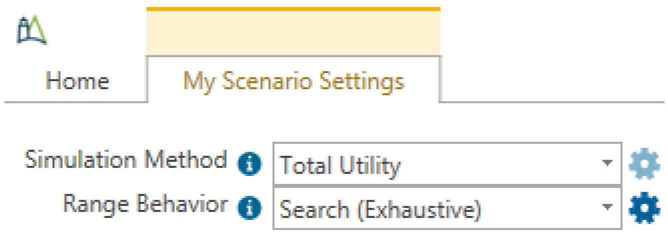


Label	Destination	Number of Nights	Hotel Quality	Hotel Type	Flight Quality	Flight Duration
Optimal	London, UK	1-3	1-3	1-2	1-3	1-3

Note that we have specified both fixed levels (for Destination) and also dynamic ranges (e.g. 1-3) for attributes. Dynamic ranges indicate the range of values that the search routine can use to maximize some criterion, yet to be specified. In the standard market simulator (without the add-on ASM capabilities), only fixed levels can be specified; there is no search capability.

This is a simple problem with just 162 possible combinations. In cases where there are few possible combinations, exhaustive search is feasible, and it *guarantees* finding the global optimum.

On the My Scenario Settings tab, we select *Total Utility* as the *Simulation Method* and select *Search (Exhaustive)* as the *Range Behavior*.



We click the Simulate button and the results are given (we show only the top three results, for brevity):

Label	Total Utility	Destination	Number of Nights	Hotel Quality	Hotel Type	Flight Quality	Flight Duration
Optimal	192.71	London, UK	8 nights	5-star hotel	Resort-type hotel	First Class accommodations on airplane (premium space & services)	Most direct flight possible

Label	Total Utility	Destination	Number of Nights	Hotel Quality	Hotel Type	Flight Quality	Flight Duration
Optimal	177.85	London, UK	6 nights	5-star hotel	Resort-type hotel	First Class accommodations on airplane (premium space & services)	Most direct flight possible

Label	Total Utility	Destination	Number of Nights	Hotel Quality	Hotel Type	Flight Quality	Flight Duration
Optimal	161.09	London, UK	8 nights	4-star hotel	Resort-type hotel	First Class accommodations on airplane (premium space & services)	Most direct flight possible

The exhaustive search tried all 162 combinations, and the total elapsed time was 1 second. The optimal product is the one we previously identified by selecting (other than London) the most preferred level from each attribute. As we calculated from the report of average utilities, the average utility for this optimal product is 192.71.

The next-best product alternatives are shown, in sort order by total utility, below the top result. The next-best product has a utility of 177.85. The only difference for this product is that the number of nights is 6 rather than 8.

Purchase Likelihood Optimizations

Searching for products that maximize Purchase Likelihood is essentially the same as a utility maximization search, except that the utility for product alternatives is transformed to a purchase likelihood scale. Purchase Likelihood simulations are proper when the part-worths have been calibrated using purchase likelihood questions from ACA or CVA. When part-worths are calibrated to purchase likelihoods, the equation below yields least-squares fit to respondents' purchase likelihood scale in the survey instrument.

$$P_x = e^x/(1+e^x)$$

The data for vacation choice came from ACA, and are scaled to produce purchase likelihood values to fit the likelihoods respondents stated during the ACA survey.

If using the Purchase Likelihood simulation with individual-level CBC data that have zero-centered logit scaling (such as with latent class or HB), the interpretation of the purchase likelihood scale is the probability of selecting this alternative relative to an alternative with mid-scale preference (utility of 0) based on the levels defined in the study.

Example #2: New Product Introduction with Competition (Share Search)

The previous section dealt with a simple—if not trivial—problem. In this example, we'll introduce the element of competition. When multiple product alternatives are considered, a share simulation is appropriate. For most share simulation cases, we'd generally recommend using the Randomized First Choice simulation method. But, simulations using Randomized First Choice take considerably longer than the other techniques. For the purposes of illustration, so you don't lose unnecessary time if following along in the software, we'll employ First Choice simulations.

For this illustration, we'll use the "TV" data set, which we often use in CBC training workshops. This is an actual data set, with 250 respondents, collected using a computerized CBC interview in the late 1990s. The subject matter is features of mid-size television sets for the home. The attributes, levels, and average part-worths (zero-centered diffs scaling) are:

Brand:

JVC	-33.72
RCA	-1.71
Sony	35.43

Screen Size:

25" screen	-29.23
26" screen	0.07
27" screen	29.16

Sound Quality:

Mono sound	-69.98
Stereo sound	24.84
Surround sound	45.14

Channel Blockout:

No channel blockout	-26.29
Channel blockout	26.29

Picture-in-picture:

No picture in picture	-32.64
Picture in picture	32.64

Price:

$300	53.19
$350	30.74
$400	-20.06
$450	-63.88

Let's assume an existing market, with the following four product offerings, and existing relative market shares (shares of First Choice):

Product #1 7.2%	**Product #2** 10.0%	**Product #3** 22.8%	**Product #4** 60.0%
JVC 25" screen Mono sound No blockout No PIP $300	JVC 26" screen Stereo sound Blockout No PIP $375	Sony 26" screen Stereo sound No blockout PIP $400	Sony 27" screen Surround sound Blockout PIP $450

We see that JVC offers a stripped-down, least expensive, model (Product #1), Sony is currently offering a full-featured expensive version (Product #4) that captures the highest share, and both JVC and Sony are offering mid-range offerings (Products #2 and #3).

Let's imagine that you represent RCA, and RCA wishes to enter this market. RCA is urging you to price the new offering at $390. You must answer two questions:

> Given the existing competition, and a price of $390, what is the optimal product to maximize share?
>
> Assuming a total market size of 1MM units sold, what is the optimal product to maximize revenue (Relative Share x Price)?

First, we'll deal with the question, "Which product can RCA offer for $390 to maximize share?" This is a tiny problem, with only 36 remaining combinations to be searched (after holding brand and price constant). Therefore, *Exhaustive Search* is appropriate.

You specify the new searched product (RCA at $390) as follows:

	Label	Brand	Screen Size	Sound Quality	Channel blockout	PIP	Price
	Product 1	JVC	25" screen	Mono sound	No channel blockout	No picture in picture	300
	Product 2	JVC	26" screen	Stereo sound	Channel blockout	No picture in picture	375
	Product 3	Sony	26" screen	Stereo sound	No channel blockout	Picture in picture	400
	Product 4	Sony	27" screen	Surround sound	Channel blockout	Picture in picture	450
	Searched	RCA	=1-3	=1-3	=1-2	=1-2	390

From the *My Scenario Settings* tab, select the *First Choice* simulation method. Select *Search (Exhaustive)* as the *Range Behavior*. Edit the settings for the Search (Exhaustive) range behavior (by clicking the settings icon) and make sure that on the Objectives dialog that *Shares of Preference* is selected.

Using Search (Exhaustive) and the First Choice simulation approach, the ASM tries all 36 possible combinations, and returns the following optimal product configuration:

RCA
27" screen
Surround sound
Blockout
PIP
$390

Share= 52.40%

We again see that the search has returned a very trivial solution. If RCA charges $390, it should provide all the best remaining features to maximize share. It is important to note that there are many studies in which attributes do not have such clearly "most preferred" levels. In such cases, the answer will no longer be so obvious.

Optimizing Revenue

The second question posed earlier was how to price an optimal RCA-branded TV to maximize overall revenue. Using the Range Behavior = Sensitivity mode with the market simulator, we can compute shares of First Choice for the "optimal" product found in the previous section at various prices. Below, we've used the =range(300,450,25) command to ask the simulator to run the simulation at prices from $300 to $500, with a step size of $25.

	Label	Brand	Screen Size	Sound Quality	Channel blockout	PIP	Price
	Product 1	JVC	25" screen	Mono sound	No channel blockout	No picture in picture	300
	Product 2	JVC	26" screen	Stereo sound	Channel blockout	No picture in picture	375
	Product 3	Sony	26" screen	Stereo sound	No channel blockout	Picture in picture	400
	Product 4	Sony	27" screen	Surround sound	Channel blockout	Picture in picture	450
	Sensitivity	RCA	27" screen	Surround sound	Channel blockout	Picture in picture	=range(300,450,25)

And, assuming a total number of units sold of 1MM to the market, we can calculate total revenue to RCA at each price point (revenue = price x share x units sold).

Price	Share	Revenue
$300	68.80%	$206,400,000
$325	66.40%	$215,800,000
$350	63.20%	$221,200,000
$375	58.00%	$217,500,000
$400	47.60%	$190,400,000
$425	36.80%	$156,400,000
$450	26.40%	$118,800,000

If evaluating every $25 increment, the highest revenue is achieved at $350. However, we can also use the ASM to search for this automatically and to report the revenue amount rather than computing it manually on your own. Furthermore, we can have the ASM investigate many more price points along the price continuum of $300 to $450, such as every $5 increment.

(Note that with this project, we already have specified that Price is a continuous variable and have associated a value of 300 with level 1, 350 with level 2, etc. We did this from the *Home* tab, *Attribute Info* icon within the *Project Information* group on the toolbar.)

To enable an optimization search where revenue is the objective function (the goal), we go to the *My Scenario Settings* tab, specify *Search (Grid)* as the *Range Behavior* and edit the settings for Search (Grid) using the settings icon). On the Objectives dialog we select Revenue as our single objective:

	Objective
☐	Shares of Preference
☑	Revenue
☐	Profit
☐	Cost

To enable searching at every $5 increment, we change the Range command for the product specifications to the following: =Range(300,450,5). Finally, to specify the market size of 1MM units sold across competing televisions, we go to the *Home* tab, click the *Revenues & Costs* icon, and specify 1000000 units within the *Market Size* dialog.

Using Grid search, we find that the optimal price is $355 for the RCA product to maximize revenue of $224,360,000. This is slightly higher than the revenue achieved earlier at $350, when we were limited to less precise $25 increments.

If we run the simulation searching at every $1 increment, we find that the optimal solution for maximizing revenue for RCA is found at a price of $359 (revenue= $225,452,000). Below the table showing the optimal results of the Grid search we find the estimated standard error for that estimate of revenue ($10,996,287). Plus or minus 1.96 standard errors describes the 95% confidence interval for the revenue estimate. We can easily see that there isn’t a statistically significant difference between this result and the total revenue of $224,360,000 at a price of $355 that we found using the Grid search and interpolation at \$5 intervals, or the earlier result when interpolating at \$25 increments.

Example #3: Searching for Multiple Products Simultaneously

For illustration, let's make the search space a bit more complex. Let's assume that RCA is interested in offering *two* televisions. What are the best two offerings to bring to the market to optimize revenue for RCA, again under the previous assumptions of competition and market size?

If we employ an interpolation step value of \$5, there are 31 unique prices to investigate over the \$300-\$450 range. With two products to search simultaneously, the total possible combinations is equal to $(3x3x2x2x31)^2 = 1,245,456$. It is often faster to organize searched multiple products so that they cover mutually exclusive search spaces, such that we avoid searching over redundant solutions where the results just reflect the same two searched products in a different order. For example, we can specify that one of the products should be below \$400 and the other should be at or above \$400:

Label	Brand	Screen Size	Sound Quality	Channel blockout	PIP	Price
Product 1	JVC	25" screen	Mono sound	No channel blockout	No picture in picture	300
Product 2	JVC	26" screen	Stereo sound	Channel blockout	No picture in picture	375
Product 3	Sony	26" screen	Stereo sound	No channel blockout	Picture in picture	400
Product 4	Sony	27" screen	Surround sound	Channel blockout	Picture in picture	450
Searched 1	RCA	=1-3	=1-3	=1-2	=1-2	=range(300,395,5)
Searched 2	RCA	=1-3	=1-3	=1-2	=1-2	=range(400,450,5)

Now, rather than 1,245,456 possible combinations to consider, the search needs to consider a fewer number: (3x3x2x2x20)(3x3x2x2x11) = 285,120. It takes about 10 minutes to run this exhaustive search. However, Grid search finds the same optimal result in seconds.

Product Search Result #1

Label	Revenue	Share of Preference	Brand	Screen Size	Sound Quality	Channel blockout	PIP	Price
Product 1	\$14,400,000.00	4.80%	JVC	25" screen	Mono sound	No channel blockout	No picture in picture	300
Product 2	\$6,000,000.00	1.60%	JVC	26" screen	Stereo sound	Channel blockout	No picture in picture	375
Product 3	\$25,600,000.00	6.40%	Sony	26" screen	Stereo sound	No channel blockout	Picture in picture	400
Product 4	\$97,200,000.00	21.60%	Sony	27" screen	Surround sound	Channel blockout	Picture in picture	450
Searched 1	\$213,000,000.00	60.00%	RCA	27" screen	Surround sound	Channel blockout	Picture in picture	355.00
Searched 2	\$22,400,000.00	5.60%	RCA	27" screen	Stereo sound	Channel blockout	Picture in picture	400.00

We can see that as with the one-product optimization solution, an RCA at about \$355 with all the best features for screen size, sound quality, channel blockout and PIP is chosen (product "Searched 1" above). In addition, a bit more revenue might be derived if the same television were simultaneously offered with the lesser quality Stereo Sound for \$400, rather than with Surround Sound. (But this doesn't seem to make much logical sense—a conundrum that can be resolved by specifying appropriate cost information and doing profitability searches, as described further below.) Recall that the best one-product simulation achieved revenue of about \$224MM, whereas the revenue with two product offering for RCA is about \$235MM.

All the simulations conducted this point have lacked the important element of cost information. In the previous example, we don't know whether RCA could stay in business manufacturing and selling these two television sets. Even though we have specified a revenue maximization strategy, RCA may very well lose money on every unit sold if the total costs for these television sets exceed the prices indicated.

As we've emphasized before, having cost information is key to obtaining the most value from product optimization searches. Without key cost data, the search results often return trivial answers that are not in the best interest of a company. In the next section, we'll extend this example by including cost information and conducting a profitability search.

A Note on Standard Errors

For ease of computation, we calculate the standard error of any net revenue, share, utility, or profit for multiple products using the formula for "pooled standard error." For example, considering products *a* and *b*, the standard error of the sum (shares, revenues, etc.) of products *a* and *b* is equal to:

$$SE_{a+b} = \text{sqrt}\ (SE_a^{\ 2} + SE_b^{\ 2})$$

Because the shares for *a* and *b* are usually negatively correlated in market simulations, this estimate provides a more conservative (slightly larger) value than if the standard error were computed on a new variable representing the net sum for *a* and *b*.

Example #4: New Product Introduction with Competition (Profitability Search)

In the previous example, RCA was interested in creating mid-sized televisions to compete with JVC and Sony, where each competitor already offered two products. We searched for an optimal configuration for the best single and pair of RCA televisions to offer, relative to the competition, to maximize Revenue. However, the fundamental weakness of the approach is that there is no guarantee that following the search recommendations is a profitable strategy for RCA.

In this example, we include the critical aspect of costs. Let's assume that RCA is able to provide a basic fixed cost structure for including various features in these mid-sized television sets:

RCA Base Cost:	$180
Screen Size:	
25" screen	+$0
26" screen	+$20
27" screen	+$35
Sound Quality:	
Mono sound	+$0
Stereo sound	+$25
Surround sound	+$40
Channel Blockout:	
No channel blockout	+$0
Channel blockout	+$15
Picture-in-picture:	
No picture in picture	+$0
Picture in picture	+$30

The ASM lets you add "cost tables" when you click *Revenues & Costs* from the *Home* tab. Click the Cost Tables dialog and then use the *Add* drop-down box to select *Add costs per attribute level…*. Cost tables let you specify costs for one attribute at a time (as with our example), or multiple attributes at a time if the costs interact with multiple attributes (e.g. if the cost for PIP depends on what screen size is offered). In our example above, there are no interdependencies for costs.

We'll assume a total of 1MM units are sold into the market, by opening the *Revenues & Costs* area from the *Home* tab, and specifying *Market Size* of 1000000. We specify profit as the single objective function (goal) from the *My Scenario Settings* dialog, by clicking the settings icon next to *Range Behavior* and clicking (only) *Profit* on the *Objectives* dialog. As per the previous example, we have specified a Range command for price that interpolates at $5 increments, and we constrained the first searched product to be below $400 and the second searched product to be $400 or more.

The optimal result (we've run this search multiple times, including via exhaustive search to confirm it) is:

Label	Profit	Share of Preference	Brand	Screen Size	Sound Quality	Channel blockout	PIP	Price
Product 1	$15,600,000.00	5.20%	JVC	25" screen	Mono sound	No channel blockout	No picture in picture	300
Product 2	$8,820,000.00	2.80%	JVC	26" screen	Stereo sound	Channel blockout	No picture in picture	375
Product 3	$29,900,000.00	9.20%	Sony	26" screen	Stereo sound	No channel blockout	Picture in picture	400
Product 4	$104,280,000.00	31.60%	Sony	27" screen	Surround sound	Channel blockout	Picture in picture	450
Searched 1	$19,200,000.00	16.00%	RCA	26" screen	Stereo sound	No channel blockout	No picture in picture	345.00
Searched 2	$40,480,000.00	35.20%	RCA	27" screen	Surround sound	Channel blockout	Picture in picture	415.00

Profit across the two RCA products is maximized at $59,680,000 ($19,200,000 + $40,480,000) by offering a full-featured TV at $415 (undercutting the price for Sony's identically full-featured TV selling for $450) and another less-fully featured TV at $345. The compromises in the less-fully featured TV are to reduce the screen size to 26", to offer Stereo rather than Surround Sound, and no channel blockout or picture in picture.

(Note: because we don't know the cost structure for the competing JVC and Sony products—we left the base price for these brands blank—the reported Profit numbers for these fixed competitors are not accurate and may be ignored.)

Example #5: Searches Using Attribute-Based Prices

As mentioned earlier, reliable cost information often is not available for conducting profitability searches. But, when utility, share, or revenue searches are conducted, the "optimal" product is often the one with the best features at the cheapest price. This is a trivial, useless result. Even though cost information may not be available, firms often have established pricing information (broken down by features, or attribute levels), and these can be incorporated into the product search. For example, whenever a certain feature is added to a product alternative, its overall price can be adjusted by a certain price delta.

Let's again refer to the TV data set that we've been using. Further, let's assume we do not have any cost information regarding the various features. Rather, we know a base price, and the incremental prices that certain features (such as picture-in-picture, or a larger screen size) add to the overall price for the unit.

Base Price:	$310	(Specify this as a fixed price for RCA products in the product specifications grid, not in the pricing tables)
Screen Size:		
25" screen	+$0	
26" screen	+$25	
27" screen	+$30	
Sound Quality:		
Mono sound	+$0	
Stereo sound	+$30	
Surround sound	+$45	
Channel Blockout:		
No channel blockout	+$0	
Channel blockout	+$20	
Picture-in-picture:		
No picture in picture	+$0	
Picture in picture	+$35	

Note that given these prices, the lowest possible price that a product can take on is $310 ($310 + $0 + $0 + $0 + $0), and the highest price possible is $440 ($310 + $30 + $45 + $20 + $35). These prices all fall within the full ranges of price that we measured, $300-$450. If some possible total prices fall outside the range, the ASM will prohibit such combinations during the search process.

In the previous two examples, RCA was interested in creating mid-sized televisions to compete with JVC and Sony, where each competitor already offered two products. We introduced the fixed competition in example #2. In example #3, we searched for an optimal configuration for the best single and pair of RCA televisions to offer, relative to the competition, to maximize Revenue. In example #4, we extended that scenario to include costs and to search based on profit. In this example, we'll use the same competitive mix, but use attribute-based pricing information instead of costs, to search for the best two products for RCA to maximize share.

To specify prices associated with attribute levels, from the *Home* tab, click *Revenues & Costs*. From the *Price Tables* dialog, click the *Add* drop-down control and select *Add prices per attribute level…*. Select all attributes to include in the pricing table *except Brand*. For each attribute level, specify the prices as shown above (do not specify a base price of $310 for the RCA brand here in the pricing tables; instead specify it as a fixed price for the RCA products in the product specification grid).

Specify that the objective (the goal) should be to optimize (only) Revenue, by clicking the settings icon next to *Range Behavior* and clicking (only) *Revenue* on the *Objectives* dialog. So that this runs quickly for the purposes of illustration, we'll continue to use the *First Choice* simulation method. The market size is assumed to be 1MM units sold across all product offerings.

The product specification grid should look like the following for this example:

Label	Brand	Screen Size	Sound Quality	Channel blockout	PIP	Price
Product 1	JVC	25" screen	Mono sound	No channel blockout	No picture in picture	300
Product 2	JVC	26" screen	Stereo sound	Channel blockout	No picture in picture	375
Product 3	Sony	26" screen	Stereo sound	No channel blockout	Picture in picture	400
Product 4	Sony	27" screen	Surround sound	Channel blockout	Picture in picture	450
Searched 1	RCA	=1-3	=1-3	=1-2	=1-2	=310
Searched 2	RCA	=1-3	=1-3	=1-2	=1-2	=310

Note that we are specifying the base price of $310 for the two RCA products here in the product specification grid. The prices for enhanced levels of screen size, sound quality, channel blockout, and PIP will be added to that base $310 as the search procedure examines products with enhanced attributes. The total price for each product will be used for determining the utility.

With this situation, there are just 1296 possible combinations to be searched $(3x3x2x2)^2$. This problem is small enough such that Exhaustive search computes in a few moments. The optimal result is:

Product Search Result #1								
Label	Revenue	Share of Preference	Brand	Screen Size	Sound Quality	Channel blockout	PIP	Price
Product 1	$16,800,000.00	5.60%	JVC	25" screen	Mono sound	No channel blockout	No picture in picture	300
Product 2	$13,500,000.00	3.60%	JVC	26" screen	Stereo sound	Channel blockout	No picture in picture	375
Product 3	$60,800,000.00	15.20%	Sony	26" screen	Stereo sound	No channel blockout	Picture in picture	400
Product 4	$169,200,000.00	37.60%	Sony	27" screen	Surround sound	Channel blockout	Picture in picture	450
Searched 1	$55,440,000.00	14.40%	RCA	27" screen	Surround sound	No channel blockout	No picture in picture	385.00
Searched 2	$103,840,000.00	23.60%	RCA	27" screen	Surround sound	Channel blockout	Picture in picture	440.00

One of the optimal RCA products ("Searched 2") has all the best features (highest screen size, sound quality, and both channel blockout and picture-in-picture) and its price is $440, which is the sum of all the attribute-based prices associated with that most feature-rich offering. In addition to that feature-rich product, RCA could capture additional share by offering a low-end model at $385, with no channel blockout or picture-in-picture.

We see that with the inclusion of attribute-based prices, the optimal result seems to reflect a more realistic, useful outcome. The optimal products for RCA to maximize share of preference are not permitted to offer the best features at the lowest prices.

Example #6: Maximize Appeal Subject to a Maximum Cost

The previous examples dealt with products sold for a price, leading to concrete measures of revenue and profits. However, there are other situations in which price, revenues and profits are either not applicable or only indirectly tied to the optimization problem at hand.

For example, one might use conjoint analysis for determining what type of health insurance plan a large company should adopt. One can enter costs into the ASM, and use these as constraints in the optimization. For example, one might seek to optimize the "Purchase Likelihood" or utility of the health plan (as perceived by the employees), subject to the cost not exceeding some threshold.

Example #7: Minimize Cost Subject to Meeting a Performance Threshold

The researcher can search for products that minimize cost, subject to meeting a certain utility, purchase likelihood, share, revenue, or profit threshold. Building upon the previous example, the problem may be to minimize the cost of a health plan such that the average desirability (purchase likelihood model) of that plan is at least 75%. Such searches can only be done using the Exhaustive search method.

Example #8: Multi-Objective Search: Maximize Both Share of Preference and Profit

Sometimes solutions that maximize share of preference do relatively poorly in terms of profit or revenue (and vice-versa). The ASM allows you to conduct multi-objective searchers that examine a variety of solutions representing Pareto-efficient tradeoffs among two or more multiple objectives. That means you can try to maximize profit without giving up much in share of preference (and vice-versa). Both exhaustive and genetic search may be used for multi-objective searches.

We will return to the setup of Example 4 (a profitability search on two products for RCA). The product specification grid looked as follows:

Label	Brand	Screen Size	Sound Quality	Channel blockout	PIP	Price
Product 1	JVC	25" screen	Mono sound	No channel blockout	No picture in picture	300
Product 2	JVC	26" screen	Stereo sound	Channel blockout	No picture in picture	375
Product 3	Sony	26" screen	Stereo sound	No channel blockout	Picture in picture	400
Product 4	Sony	27" screen	Surround sound	Channel blockout	Picture in picture	450
Searched 1	RCA	=1-3	=1-3	=1-2	=1-2	=range(300,395,5)
Searched 2	RCA	=1-3	=1-3	=1-2	=1-2	=range(400,450,5)

Costs were specified per level as earlier shown in Example 4. No attribute-based price tables were specified (so if you have price tables specified, you should delete this for this example).

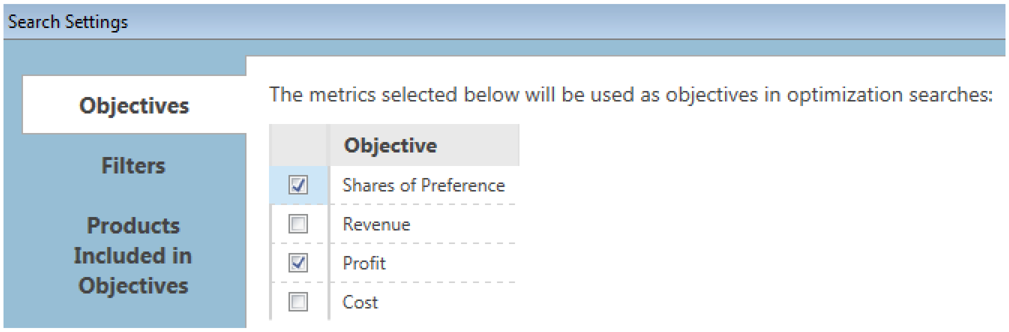
Using Grid Search, we previous found a profit-maximizing solution for RCA as follows:

Label	Profit	Share of Preference	Brand	Screen Size	Sound Quality	Channel blockout	PIP	Price
Product 1	$15,600,000.00	5.20%	JVC	25" screen	Mono sound	No channel blockout	No picture in picture	300
Product 2	$8,820,000.00	2.80%	JVC	26" screen	Stereo sound	Channel blockout	No picture in picture	375
Product 3	$29,900,000.00	9.20%	Sony	26" screen	Stereo sound	No channel blockout	Picture in picture	400
Product 4	$104,280,000.00	31.60%	Sony	27" screen	Surround sound	Channel blockout	Picture in picture	450
Searched 1	$19,200,000.00	16.00%	RCA	26" screen	Stereo sound	No channel blockout	No picture in picture	345.00
Searched 2	$40,480,000.00	35.20%	RCA	27" screen	Surround sound	Channel blockout	Picture in picture	415.00

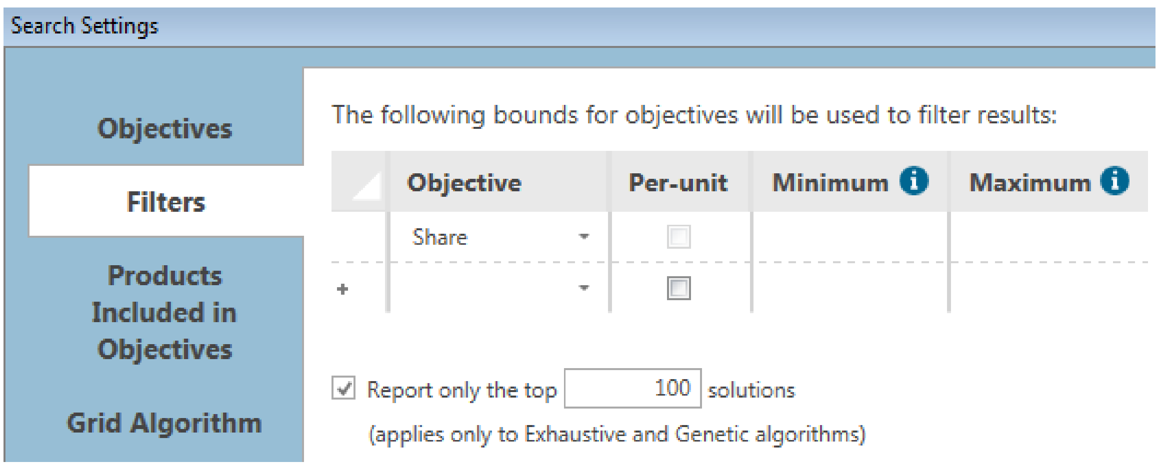
Net profit for the two RCA products was $19,200,000 + $40,480,000 = $59,680,000 with a net share of preference to RCA of 16.00% + 35.20% = 51.20%.

With multi-objective search, the question is whether a different solution could trade away not much profit yet achieve significantly higher share of preference.

To conduct a multi-objective search, from the *My Scenario Settings* tab we specify either *Exhaustive Search* or *Genetic Search* as the *Range Behavior* and on the Settings icon next to *Range Behavior* we click both objectives (Share of Preference & Profit):



Let's use *Exhaustive Search* and only examine the top 100 solutions that offer the best tradeoffs among Share of Preference and Profit. To limit the reported results to the top 100 solutions, go to the *Filters* dialog and specify to only include the top 100 solutions:



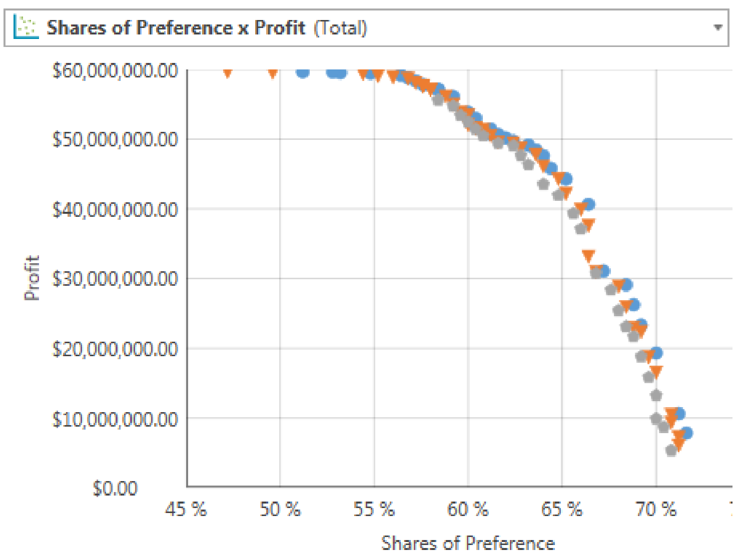
There are 285,110 possible combinations of these two RCA products to search. For speed of illustration for this white paper, we're using the First Choice method which takes about 5 minutes to complete for this exhaustive optimization search (the faster genetic search could also be used for multi-objective search).

The ASM examines the Share of Preference and Profit for each solution and returns a table report along with an interactive graphic that focuses on the top 100 solutions (as we requested) offering the most efficient tradeoffs among Share of Preference and Profit.

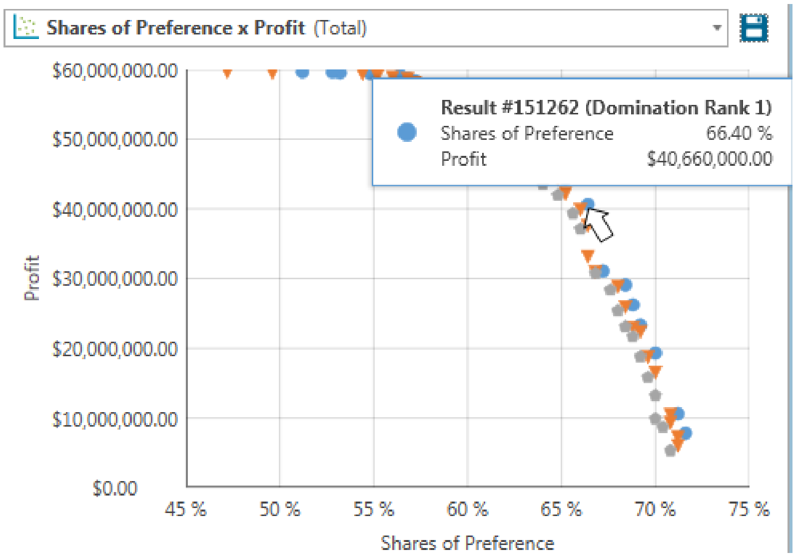
The table report sorts the top 100 solutions (providing the most efficient tradeoff of share of preference vs. profit) from highest share of preference to lowest share of preference (top few solutions shown below). Notice the highest share of preference (71.60%) is associated with a relatively low profit of $7,820,000.

	A	B	C	D
1	**Simulation Results**			
2				
3	**Summary**			
4				
5	**Domination Rank**	**Result Index**	**Share of Preference**	**Profit**
6	1	277542	71.60%	$7,820,000.00
7	1	277938	71.20%	$10,600,000.00
8	1	279126	70.00%	$19,320,000.00
9	1	245906	69.20%	$23,300,000.00
10	1	246302	68.80%	$26,240,000.00
11	1	246698	68.40%	$29,100,000.00
12	1	247094	67.20%	$31,080,000.00
13	1	151262	66.40%	$40,660,000.00

A graphic is displayed next to the table with the 100 best tradeoffs among the two objectives (Rank Order 1 solutions, where no other solution is better than or equal to it on both dimensions) displayed as blue dots (orange triangle solutions as Rank Order 2 solutions, and grey pentagons as Rank Order 3 solutions):



You can hover your mouse over any icon (solution) on the graphic to learn its identification and product specifications. For example, imagine we wanted to examine the details for a solution on the Pareto efficient frontier offering a profit around $40,000,000:



When you hover your mouse over the solution that interests you, a pop-up tells you which solution number it is (#151262, so you can easily search for the details of that solution in the table output). Here are the results for solution $151262 in the table report:

Product Search Result #151262 (Domination Rank 1)								
Label	Share of Preference	Profit	Brand	Screen Size	Sound Quality	Channel blockout	PIP	Price
Product 1	2.00%	$6,000,000.00	JVC	25" screen	Mono sound	No channel blockout	No picture in picture	300
Product 2	1.60%	$5,040,000.00	JVC	26" screen	Stereo sound	Channel blockout	No picture in picture	375
Product 3	6.80%	$22,100,000.00	Sony	26" screen	Stereo sound	No channel blockout	Picture in picture	400
Product 4	23.20%	$76,560,000.00	Sony	27" screen	Surround sound	Channel blockout	Picture in picture	450
Searched 1	39.60%	$13,860,000.00	RCA	26" screen	Stereo sound	Channel blockout	Picture in picture	305.00
Searched 2	26.80%	$26,800,000.00	RCA	27" screen	Surround sound	Channel blockout	Picture in picture	400.00

Compared to the optimal profit solution with $59,680,000 that had a net share of preference to RCA of 51.20%, the multi-objective solution has a profit of $40,660,000 and a net share of preference of 66.40%. It offers a gain of 15.2% points in share of preference for a loss of essentially $19MM in profit.

Obviously other solutions could be investigated that fit RCA's goals and provided solutions that made sense from a manufacturing and marketing standpoint to produce. For example, a solution that gives up very little in terms of profit $59,220,000 relative to the profit-maximizing solution ($59,680,000) offers a modest increase in share of preference to 56.40% (compared to 51.20%).

References

Balakrishnan, P. V. and Varghese S. Jacob (1996), "Genetic Algorithms for Product Design.," *Management Science*, 42 (8),1105-1117.

Green, Paul E. and V. R. Rao (1971), "Conjoint Measurement for Quantifying Judgmental Data," *Journal of Marketing Research*, 8 (August), 355-363.

Green, Paul E. and Abba M. Krieger (1993), "Conjoint Analysis with Product Positioning Applications", Handbooks in Operations Research and Management Sciences", Volume 5, J. Eliashberg and G. L. Lilien (eds.). Amsterdam: North-Holland, Chapter 10, pp. 467-515.

Orme, Bryan K. (2018), "Multi-Objective Searches in Conjoint Analysis: An Intuitive Explanation," accessed at: http://www.sawtoothsoftware.com/download/techpap/MultiObjectiveSearches2018.pdf.