A Multi-Objective Evolutionary Algorithm for Multivariate Optimal Allocation

Charles D. Day
Internal Revenue Service, Statistics of Income Division

KEY WORDS: Genetic Algorithm, Stratified Sampling, Evolutionary Algorithm, Convex Optimization

INTRODUCTION

When a statistician designs a stratified sample he or she must determine the allocation of the available budget for sample units to the strata. When a population statistic will be estimated for a single quantity, or on several well-correlated quantities, methods for doing this are straightforward. More often, a survey practitioner wishes to make estimates for many quantities from a single survey. In that case, optimal allocation of the sample units to strata is not so simple. Such a problem has multiple objectives, minimizing cost as well as the variances of each of the estimates of interest.

Traditionally, this problem is solved by converting the multiple objective functions into a single scalar-valued objective in one of two ways. One method involves creating a function, such as a linear combination, from the multiple objective function values and minimizing this function. A second method involves choosing one objective (usually cost minimization) and turning the rest of the objectives into constraints by setting maximum acceptable variances for each estimate of interest. Several authors have published such methods [1-8].

This paper offers an alternative. Rather than specifying a function of the objectives *a priori*, or choosing a set of arbitrary variance targets, a multi-objective evolutionary algorithm is used to generate multiple solutions that, taken together, describe the *Pareto front* for the problem. The Pareto front consists of a set of non-dominated solutions. A solution is said to be dominated if another solution exists that is better on at least one objective and at least as good on all the other objectives. A solution that is not dominated by any other solution is said to be non-dominated. Examination of the Pareto front makes the trade-offs implied by the choice of function parameters or variance targets in the traditional methods explicit.

This paper will describe the multivariate optimal allocation problem. It will then cover the basics of single-objective evolutionary algorithms, and extend these to describe a multi-objective evolutionary algorithm. The paper then offers details of the application of the multi-objective algorithm to a multivariate optimal allocation problem from the literature.

MULTIVARIATE OPTIMAL ALLOCATION

One goal of stratified sampling is to increase the precision (reduce the variance) of estimates of population statistics inferred from a sample. All other things being equal, increased homogeneity of the population being sampled works to increase precision. By dividing the population of interest into non-overlapping subpopulations (sampling strata) that are more nearly homogeneous, selecting independent samples from each stratum, and combining estimates from the strata, the statistician can make a more precise estimate than by choosing a random sample from the population as a whole.

Once stratum boundaries have been defined, the problem arises of how many sample units to allocate to each stratum. If the survey practitioner wishes only to make as precise as possible an estimate for one variable given a fixed cost, or find the minimum cost design to achieve a target variance, this problem has a well-known solution [1]:

$$n_h = n \frac{N_h S_h / \sqrt{c_h}}{\sum (N_h S_h / \sqrt{c_h})}$$

where n_h is the number of sample units allocated to stratum h, N_h is the number of population units in stratum h, c_h is the cost per unit in stratum h, S_h is the population standard deviation for the variable of interest in stratum h, and n is the total sample size. (S_h is usually estimated from frame information or earlier samples.) If a target variance is fixed and cost is to be minimized, then:

$$n = \frac{(\sum W_h S_h \sqrt{c_h}) \sum W_h S_h / \sqrt{c_h}}{V + (1/N) \sum W_h S_h^2}$$

where $W = N_h/N$. If cost is fixed and variance is to be minimized then:

$$n = \frac{(C - c_0) \sum N_h S_h / \sqrt{c_h}}{\sum N_h S_h \sqrt{c_h}}.$$

While it is rarely the case that a survey is conducted to estimate the value of only one variable, this formula is still broadly useful, since an allocation that is optimal for one variable may be near-optimal for variables that are strongly correlated with it. If, however, precise estimates of several variables are needed, and those variables are not all highly correlated with each other, it is desirable to have a method to find a good compromise allocation that will give adequate precision for all of the variables of interest. This is the usual goal of multivariate optimal allocation.

There are two common ways to approach this problem. One is to minimize a weighted sum of the variances of the variables of interest. Khan and Ahsan [7] propose a method in which they formulate this problem as a nonlinear programming problem and use a dynamic programming technique to find a solution. One problem with this approach is how to weight the variances. There is no single solution for doing this, and it is not always easy to predict what the consequences of a particular choice of weights are. Even examination of different sets of weights for the variances may not give a representative idea of the trade-offs being made due to non-linearity of the relation between the weight vector and the vectors of values of the multiple objective functions.

The other approach is to choose an acceptable coefficient of variation for each of the variables on which the allocation is to be done. These become constraints on a cost function that can be minimized, giving the following convex programming problem:

Min: $\sum c_h n_h$

s.t. $\sum_{h=1}^{H} W_h^2 S_{hj}^2 / n_h t_j^2 \overline{Y}_j^2 \le 1$ for every j

$$n > 0$$

Where t_j is the target coefficient of variation (CV) of the jth variable and $\bar{Y}_j$ is the population mean of jth variable [14]. These approaches turn the multi-objective optimization problem of multivariate optimal allocation into a problem with a single scalar-valued objective. By doing so, they allow traditional methods for solving optimization problems to be used.

EVOLUTIONARY ALGORITHMS

Briefly, single-objective evolutionary algorithms (EAs) adopt biological evolution as a model for computing. While there are a number of canonical variants of evolutionary algorithms, it is common for practitioners to adapt features of two or more variants to develop algorithms specific to the solution of their problems.

In general, evolutionary algorithms start with a "population." Each individual in the population consists of one candidate solution for the problem the EA is trying to solve. Borrowing terminology from biology, each variable in a solution is referred to as a gene, the value for each gene is called an allele, and the structure of the whole solution is referred to as a genome. These candidate solutions are usually generated at random from the space (or a well-chosen subspace) of all possible solutions.

The "fitness" of each individual is then evaluated; that is, the value of the objective function of the optimization problem being solved is determined for each candidate solution. Next, pairs (or n-tuples, should the practitioner wish) of individuals are selected to "reproduce." This selection is done in such a way as to favor fitter individuals; for example, individuals could be selected with probability proportional to their fitnesses.

During reproduction, two operations can be used to produce "children" (the next "generation" of candidate solutions). One consists of taking one part of one of the individuals selected to reproduce and appending it to the complementary part of the individual it was paired with during selection. This is referred to as "crossover" in the EA literature, and is analogous to recombination in biological reproduction (Figure 1)..

The second reproductive operator is mutation. As one might

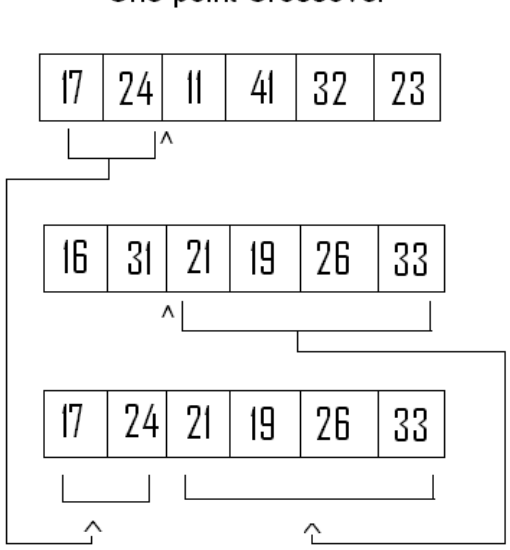


Figure 1. An example of one-point crossover.

suspect, it consists in changing the value of one of the genes with some probability. Following reproduction, each child's fitness is assessed. Children are allowed to survive into the next generation (where they become the initial population) based on their fitnesses.

This process continues, with the children becoming the next generation's parents, until some convergence criterion is reached, or a maximum number of generations is reached. One problem with EAs as described to this point is that the best solution may be lost; that is, the solution with the overall highest (if maximizing) or lowest (if minimizing) value of the objective function may disappear as the algorithm moves from generation to generation, never to be seen again. To address this problem, practitioners usually employ "elitism," allowing the k highest valued members of the current population to survive into the next generation.

Should the reader wish a thorough introduction to the field of Evolutionary Algorithms, De Jong [9] provides one.

A MULTI-OBJECTIVE EVOLUTIONARY ALGORITHM: THE STRENGTH PARETO EVOLUTIONARY ALGORITHM (SPEA2)

As indicated in the introduction to this paper, it is not necessary to use one of the common methods of creating a scalar-valued objective function from the values of multiple objective functions, or to turn all but one of the multiple objectives in a multi-objective optimization problem such as multivariate optimal allocation into constraints. In fact, algorithms are available to generate the Pareto front described in the introduction. Evolutionary algorithms, because they are already generating populations of solutions, are a natural choice for such a method. The investigation described in this paper used a multi-objective evolutionary algorithm, the Strength Pareto Evolutionary Algorithm

(SPEA2) [10] to find the Pareto front for the example problem from Bethel [6] as corrected in Zayatz and Sigman [11].

In addition to the population of candidate solutions a single-objective EA uses, SPEA2 has an *archive*. The archive contains all of the currently discovered non-dominated solutions to the problem the algorithm is attempting to solve. The archive has a maximum size, and, should it become filled, the fittest non-dominated solutions are kept and any less fit solutions beyond the maximum archive size are discarded.

In its first generation SPEA2 initializes its population. In each generation, it then stores any non-dominated solutions (considering both the archive and population) in the archive, subject to the maximum size limitation. Fitnesses are then assigned to individuals in both the population and the archive according to a method described below. SPEA2 then uses binary tournament selection to choose partners for reproductive selection from the archive.

The phrase "tournament selection" needs a little explanation. In any EA, it is necessary to select individuals for crossover and mutation. These selections are done in such a way that fitter individuals have a greater chance of being chosen to reproduce, thus putting pressure on the whole system to evolve fitter and fitter individuals in successive generations. One method of choosing individuals to reproduce is to hold a "tournament" in which k individuals are chosen at random and the fittest is selected to reproduce. The larger k gets, the more likely the randomly chosen participants are to contain at least one high fitness individual; therefore, larger values of k are associated with greater selection pressure. In binary tournament selection, the value of $k = 2$.

Recombination and mutation steps, as described for single-objective EAs, are then performed to create the next generation. This generation is the new initial population and the algorithm begins at the first step again, continuing until some convergence criterion is reached or a maximum number of generations has been reached.

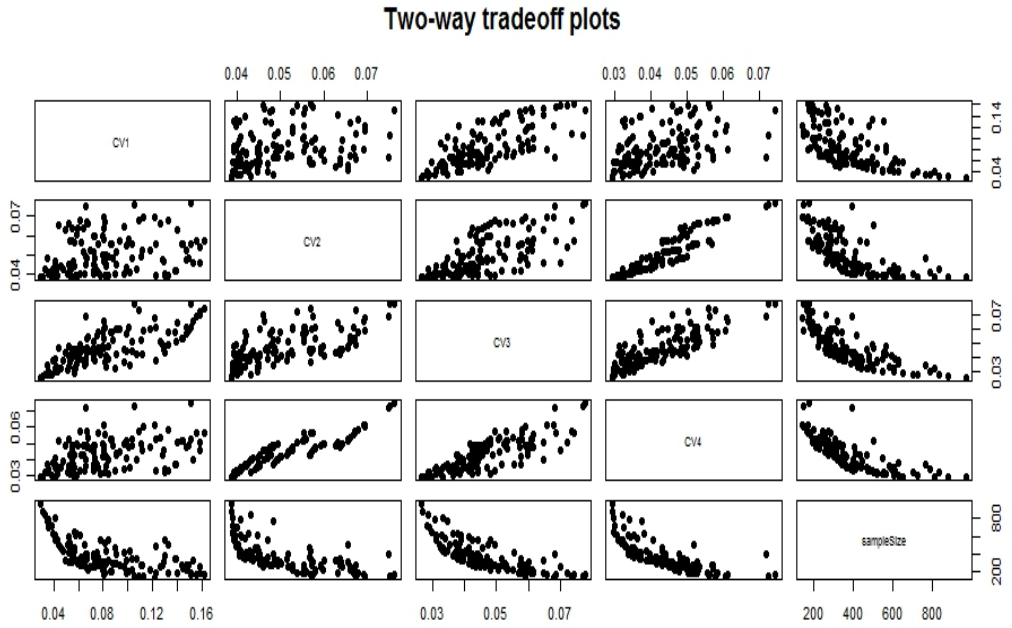
The fitness calculation for SPEA2 is unlike the calculation for single-objective EAs and is performed on the population and archive together. First, an individual is assigned a *strength* equal to the number of individuals it dominates. Next, it is assigned a *raw fitness* equal to the sum of the strengths of individuals that dominate it. Note that lower raw fitnesses are better, and that a non-dominated individual has a raw fitness of 0. If the raw fitnesses were used alone, then the archive would tend to converge to a single solution. The goal of a multi-objective EA is to produce a representative sample of non-dominated solutions. This requires the introduction of some other measure to ensure that diversity is maintained in the archive. The measure used in SPEA2 is called the *density*. The Euclidean distance between the individual being evaluated its kth (k is usually the square root of the sum of the population and archive sizes) nearest neighbor in the space of objective function value vectors is calculated, two added to it to guarantee a positive value, and the density is calculated as the inverse of the sum. The density is then added to the raw fitness to produce the final fitness value.

DESIGN OF A MULTI-OBJECTIVE EA TO SOLVE A MULTIVARIATE OPTIMAL ALLOCATION PROBLEM

This paper demonstrates the application of the SPEA2 algorithm to Bethel's multivariate optimal allocation problem [6, 11]. To solve any problem with an EA, we first must find a representation for a solution. Bethel's problem has four variables of interest and six strata in its design. The easiest representation for an allocation to six strata is simply to use a vector of six integers. As a practical constraint, each integer value was allowed to vary from 2 to 200, indicating how may sample units were assigned to each stratum. The objective function value vectors contained five floating point values, one for the budget (sum of the units allocated to the strata, assuming unit cost in each stratum), and one each for the coefficient of variation for each variable of interest, all of which are to be minimized. All objective values were normalized to lie in a range between zero and one. A population size of 1,000 was used, and an archive size of 125.

RESULTS

Results, graphed in Figure 1, were quite encouraging. Figure 1 is a scatterplot matrix that compares all of the two-way trade-offs between values of pairs of objectives for the solutions on the Pareto front. For example, the center plot in the first row displays the tradeoff between values of the CV for the third variable of interest with values of the CV for the first variable of interest. There appears to be some linear relationship between the two.



Several features of the plot deserve mention. First, the last row displays the relationship between sample size and the CVs for the four variables of interest. The plots show, as one would expect, that as sample size increases, the CVs decrease. In a single objective simple random sample design, as sample sizes increase, one would expect a decrease in CV proportional to the square root of the sample size. Such a decrease would produce a smooth curve, but three of the four CV versus sample size plots have distinct inflection points, at which the return to increased sample size in terms of reduced CVs becomes considerably smaller. A survey designer interested in minimizing total survey error might use the knowledge of where those inflection points lie to choose when to devote resources to other parts of the survey process (for example, data editing or questionnaire pre-testing) instead of increasing sample size.

The relationships of the CVs to each other also reveal interesting features of the problem. The author has previous experience with the solution of this problem [12]. Note the apparent linear relationships between CVs for variables one and three, one and four, two and three, two and four, and three and four. Note further the apparent lack of a relationship between CVs for variables one and two. When solving the problem using cost as an objective and target CV constraints, only the CV constraints for variables one and two were binding. The cause for this is apparent from the relationships revealed in the scatterplot matrix. Perhaps a survey designer might simplify his or her optimization problem by dropping the CVs for variables three and four, allowing them to "come along for the ride" with variables one and two.

CONCLUSIONS

The use of multi-objective EAs for multivariate optimal allocation holds considerable promise. Not only can they allow the survey designer to allocate his or her resources to different parts of the survey process in a more informed way, but they can also illuminate relationships between different objectives. Future research might include the application of similar methods to optimal allocation of resources across the entire survey process.

ACKNOWLEDGEMENTS

The author thanks the Statistics of Income Division, Internal Revenue Service for supporting this work. In addition, a debt of thanks is owed to Sean Luke and a number of graduate students in the Computer Science Department at George Mason University for their development of the Evolutionary Computation in Java (ECJ) library [13], which was used to develop the programs employed in this work.

REFERENCES

[1] Cochran, W. G. *Sampling Techniques*, 3rd edition, John Wiley and Sons, New York, NY, 1977, pp. 97-98.

[2] Kokan, A. R. and Khan, S. (1967) Optimum Allocation in Multivariate Surveys: An Analytical Solution. *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 29, pp. 115-125

[3] Huddleston, H. F., Claypool, P. L., and Hocking, R. R. (1970) Optimal Sample Allocation to Strata Using Convex Programming. *Applied Statistics*, Vol. 19, pp. 273-278.

[4] Kish, L. (1976) Optima and Proxima in Linear Sample Designs. *Journal of the Royal Statistical Society. Series A*, Vol. 159, pp. 80-95

[5] Chromy, J. B. Design Optimization With Multiple Objectives. In *Proceedings of the Section on Survey Research Methods, 1987*. American Statistical Association, pp. 194-199.

[6] Bethel, J. (1989) Sample Allocation in Multivariate Surveys. *Survey Methodology*, Vol. 15, pp. 47-57.

[7] Khan, M. G. M. and Ahsan, M. J. A Note on Optimum Allocation in Multivariate Stratified Sampling. *South Pacific Journal of Natural Science*, Vol. 21, pp. 91-95.

[8] Winkler, W. E. (2004) Sample Allocation and Stratification. Chapter in *Handbook of Sampling Techniques and Analysis*, unpublished.

[9] DeJong, K. A. (2006) *Evolutionary Computation: A Unified Approach.* MIT Press, Boston, MA.

[10] Zitzler, E., Laumanns, M., and Thiele, L. (2001) SPEA2: Improving the Strength Pareto Evolutionary Algorithm. TIK-Report 103, Swiss Federal Institute of Technology, Zurich, Switzerland.

[11] Zayatz,, L. and Sigman, R. (1995) CHROMY_GEN: General Purpose Program for Multivariate Allocation of Stratified Samples Using Chromy's Algorithm." Economics Statistical Reports Series ESM-9502, Washington, DC: Bureau of the Census.

[12] Day, C. Application of an Evolutionary Algorithm to Multivariate Optimal Allocation in Stratified Sample Designs. *2006 Proceedings of the American Statistical Association*, Survey Research Methods Section [CD-ROM]. Alexandria, VA: American Statistical Association

[13] Luke, S. et al. Evolutionary Computation in Java (ECJ), *http://www.cs.gmu.edu/~eclab/projects/ecj/*.