APPLICATION OF AN EVOLUTIONARY ALGORITHM TO MULTIVARIATE OPTIMAL ALLOCATION IN STRATIFIED SAMPLE DESIGNS

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

KEY WORDS: Genetic Algorithm, Stochastic Search

INTRODUCTION

Biological evolution can be viewed as a process of optimizing a species to (or increasing its fitness for) its environment. Evolutionary Algorithms (EAs), sometimes called Genetic Algorithms after their most common variant, adopt biological evolution as a model for computation. These algorithms are used most often for finding approximate solutions to computationally intractable optimization problems. In this paper, an evolutionary algorithm is applied to the problem of multivariate optimal allocation in stratified sample designs.

The work reported on in this paper focuses on the design of an EA for solving the multivariate optimal allocation problem and an investigation of the performance of that algorithm on a simple, well-known example. One of the most attractive features of EAs is the flexibility of their "fitness" (or objective) functions. Many characteristics can be optimized simultaneously. Future research will explore how an EA might be used to find the optimal strata boundaries and the optimal allocation of sample units to strata simultaneously, with the goal of producing a better result than doing so in serial fashion using standard methods

Stratified sample designs are employed for several reasons. These include: 1) to increase the precision of estimates for the whole population for one or more key data items being collected in the survey; 2) to obtain more precise estimates for interesting domains; 3) to allow the use of different sampling, nonresponse adjustment, editing, or estimation methods for domains with differing characteristics affecting the choice of method, and 4) to facilitate administration of the survey [1]. This paper focuses on the first two reasons.

Once stratified sampling has been chosen, it is necessary to determine how to divide the population into strata, and how to allocate the sample to those strata. One decision that must be made is the choice of a variable or variables on which to stratify. Since it is rare to conduct a survey with only one item of interest, the stratification variable or variables are chosen (or constructed) to have a strong correlation with as many items of interest as possible. Methods for construction of optimal stratum boundaries (with the goal of improving the precision of estimates) have been proposed by Dalenius and Hodges [2], Singh [3], Lavallée and Hidiroglou [4], and Sweet and Sigman [5].

Once stratum boundaries have been defined, and a maximum sample size or total cost determined, it is straightforward to determine the number of sample units to allocate to each stratum if the allocation is done on a single variable [6]. The problem becomes more difficult if the allocation is done on multiple variables. A number of approaches have been use to find good approximations to the optimum allocation [7-11]. This paper proposes the use of another method.

EVOLUTIONARY ALGORITHMS

As described in the introduction, evolutionary algorithms 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. For example, if an EA were designed to find the rational roots of a quadratic equation, the solutions might be represented by a vector in $\mathbf{Q}^2$ (a vector of two floating point numbers). The genome would be

a vector of two floating pointing numbers, each of the two variables would be a gene, and the value assigned to each variable an allele. The representation of candidate solutions is an important factor in the success of an EA; therefore, representations must be chosen with care.

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 individual. Note that the objective function can be as complicated as a simulation for flow of a gas or liquid through a manifold or as simple as a single polynomial, so long as it is possible to rank the candidate solutions on their fitnesses.

Next, pairs (or n-tuples, should the practitioner wish) of individuals are selected to "reproduce." This selection is done proportionate to the individuals' fitness. How the fitnesses are weighted in determining the probability of an individual's selection to reproduce is one of, as Kenneth DeJong calls them, the "knobs" that one has to turn in tuning an EA for optimal performance. If fitter individuals are given a great deal higher probability of selection than those that are less fit, then the EA is expected to converge more quickly to an answer, but at greater risk of finding a local, rather than the global, optimum. The less "selection pressure" is applied, the more fully the EA is allowed to explore the solution space, at the cost of slower convergence and at the risk of not converging at all. This trade-off is referred to as "exploitation versus exploration," and a well-designed EA must balance the two competing goals so that progress is made toward convergence without the EA getting stuck in a local optimum.

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. Given possible constraints on the structure of solutions, the design of crossover operators can become quite creative. The desire for simpler or more effective crossover operators can also impact the representation of solutions.

The second reproductive operator is mutation. As one might suspect, it consists in changing the value of one of the genes with some probability. Similarly to selection pressure, if the mutation rate (the probability of a mutation) is high, the EA will be expected to more fully explore the solution space, if it is lower, convergence is expected to occur more quickly.

Following reproduction, each child's fitness is assessed. Children are allowed to survive into the next generation (where they become the initial population) in proportion to their fitness. The earlier comments about selection pressure apply to survival selection as well as they do to reproductive selection.

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.

THE MULTIVARIATE OPTIMAL ALLOCATION PROBLEM

In stratified sampling, 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 [12]:

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

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 find 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 [13] 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.

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 $\overline{Y}_j$ is the population mean of jth variable [14].

DESIGN OF AN EA TO SOLVE THE MULTIVARIATE OPTIMAL ALLOCATION PROBLEM

When designing an EA (or any other optimization algorithm), it is important to incorporate any special features of the problem to be solved. The multivariate optimal allocation problem has two features that should be accounted for in the design. First, it is really two problems, minimize a function of variances for a fixed cost or minimize cost subject to fixed variance targets. A good solution will allow the statistician to choose which of these approaches to follow.

Second, any solution that results in enough of the available budget (total cost) being left over to allocate another unit in any stratum is sub-optimal; that is, any optimal solution must use the entire budget. This implies that, rather than searching the entire space of feasible allocations, the EA can concentrate on a bounding hyperplane of that space, enormously reducing the size of the set to be searched and reducing time to convergence.

In order to produce a method that can use either the fixed cost or fixed variance targets approach, a decision was made to design an EA that was a framework for optimization. This framework searches for a solution that meets a vector of target CVs with a fixed budget and terminates either when a solution is found or when a maximum number of generations is reached. The EA framework is then embedded in a program that allows the user to find the minimum cost for fixed variance targets or the minimum variances for a fixed budget (total cost) using a binary search approach. Note that the method can be adapted so that the variance targets in the second approach can be searched with any priority scheme the user desires.

Candidate solutions to the multivariate optimal allocation problem (individuals) are represented as integer vectors of length H, where H is the number of strata. Each element of the vector (gene) is assigned a fraction of the total budget for the sample [15]. The stratum budget is chosen such that it is divisible by the cost of a unit in that stratum.

Given a vector of allocations to strata, the program calculates a “standardized precision unit” (SPU) [16] for each variable *j* as follows:

$$SPU = \sum_{i=1}^{H} W_i^2 S_{ij}^2 / n_i t_j^2 \overline{Y}_j^2$$

Note that this is the left-hand-side of the first constraint on the cost minimization problem discussed earlier. Further, the variance constraint on the jth variable is met when this quantity is equal to one. Using the SPUs a fitness function can be formed. This EA uses:

$$fitness = \sum_{SPU_j > 1} SPU_j^2$$

as its fitness function. This function is minimized.

The EA is designed to constrain the search to the bounding hyperplane where the optimum solution lies by embedding this constraint in the initialization, mutation, and crossover operators. When creating an initial population of candidate solutions, the initialization operator first chooses one gene at random with uniform probability. The operator then assigns some fraction (again, chosen at random) of the budget to that gene as its value, taking care to always leave enough budget so that no gene is initially assigned a budget less than the cost of two units, or greater than the budget required to sample the stratum at 100 percent. The operator then proceeds to fill in the rest of the vector in a similar fashion, resulting in a vector that uses the whole budget without exceeding it. So, the EA starts with an initial population of vectors that lie in the hyperplane with the optimal solution.

If the search is to be constrained to that hyperplane, mutation and crossover operators must operate to keep children in that region of the solution space as well. The mutation operator uses a parameter that contains the probability of mutation to decide whether or not to mutate a particular gene. If a gene is chosen for mutation, its value is increased or decreased (with equal probability) by the cost of one unit in that stratum, subject to the constraints that no stratum is allocated fewer than two nor more than N_h units. Another gene is chosen at random (with equal probability) to be adjusted to maintain the constraint that the whole budget is used and not exceeded. Designing a crossover operator was more challenging. In simple situations, one-point crossover (which was used in this EA) simply takes one parent's gene values prior to the crossover point and appends the other parent's gene values after that point. Clearly, this would not guarantee that the constraint that the whole budget must be used but not exceeded would be met. A proportional crossover operator was designed instead. This operator created a child by taking one parent's values prior to the crossover point, and allocating the remaining budget according to the proportion of the remaining budget after the crossover point in the second parent assigned to each gene in the second parent. For example, if there were two parents in a problem with n = 60, (25, 21, 10, 4) and (10, 15, 15, 20), and the randomly chosen crossover point was after gene 2, then the first child would have its first two values equal (25, 21, …). The last two values would be determined as follows. A budget of 14 (60 – (25 + 21)) remains to be allocated. The second parent's last two genes have values in the proportion 3:4. Allocating the remaining budget of 14 in the proportion 3:4 results in a child with the allocation (25, 21, 6, 8).

Selection for reproduction is done using the roulette-wheel method. This method selects an individual *i* from the population with probability

$$P(selection_i) = fitness_i / \sum_{k=1}^{m} fitness_k$$

Where m is the number of individuals in the population. Survival selection will also be done using the roulette-wheel method. As in most implementations, elitism is employed to avoid losing the best solution found to that point.

PERFORMANCE ON AN EXAMPLE PROBLEM

In order the test the performance of the EA framework, a simple problem from the literature

Table 1. Results from Runs 1 through 10 of EA Framework on Bethel's Problem

Run No.	Allocation						CVs			
1	90	31	28	39	33	20	.060	.060	.047	.050
2	90	31	29	40	34	17	.060	.060	.048	.050
3	89	32	29	40	32	19	.060	.060	.047	.050
4	91	28	30	40	33	19	.060	.060	.048	.050
5	92	28	23	45	36	17	.060	.060	.048	.050
6	92	25	27	47	33	17	.060	.060	.049	.051
7	89	33	27	43	32	17	.060	.060	.047	.050
8	91	29	24	40	39	18	.060	.060	.048	.050
9	92	25	28	43	35	18	.060	.060	.049	.050
10	92	25	28	44	33	19	.060	.060	.049	.050

was chosen. This problem is well-described in Bethel's 1987 paper [17]. A correction to a misprinted value in the problem can be found in Zayatz and Sigman [18]. The problem involves an allocation to six strata based on four variables. Bethel finds a minimum cost of 241 units to meet a desired CV of 0.06 for each of the four variables. (All strata have equal unit costs.) Bethel found the solution (90, 29, 27, 43, 34, 18) with resulting CVs of 0.060, 0.060, 0.048, and 0.050. Note that the last two variables' target CVs are not binding constraints.

Results from ten runs of the EA framework with same target CVs and a budget of 241 are contained in Table 1.

Given the same budget and variance constraints, the EA found ten different solutions, all of which produced CVs that were the same or very nearly the same as those found by Bethel. That a number of different solutions exist to an optimization problem that involves minimization of a convex function should be no surprise.

Convergence properties of the algorithm were remarkably good. The algorithm never failed to find the optimum solution in less than 5,000 generations, and in only one run did it require more than 500. This required a few seconds on a slow computer. It is extremely rare to find an EA that converges in so few generations; hundreds of thousands or millions of generations are more the norm. It will be interesting to see if this performance holds up when larger, more difficult problems are attempted.

As a demonstration of the use of the framework to actually find an optimal solution, a program was run that conducted a binary search for the minimum cost solution to Bethel's problem with the stated target CVs, using the EA to test whether a solution could be found with a given budget in 5,000 generations. Using upper and lower bounds of 300 and 0 for the range of the cost search, the program found a solution using 240 units in seven iterations. (The cost was one unit less than Bethel's due to his rounding a real-valued solution to integers, while this method solves for an integer solution directly.)

CONCLUSIONS AND FUTURE WORK

An Evolutionary Algorithm can be used to solve the multivariate optimal allocation problem. Results are similar to other methods. The real promise of this technique lies in extensions to more complicated problems. Today, optimal stratum boundaries and optimal allocations given those boundaries are found separately. With a more complicated representation, it should be possible to solve for optimal stratum boundaries and multivariate optimal allocations to those strata simultaneously. In addition, more complicated representations and objective functions could be used to solve for allocations on criteria other than minimizing variance or cost, such as inclusion of a sufficient number of units with a particular condition to allow good modeling results from a sample.

ACKNOWLEDGEMENTS

The author thanks the Statistics of Income Division, Internal Revenue Service for supporting this work. In particular, he would like to thank Yahia Ahmed for sharing his expertise in sampling and for his encouragement.

REFERENCES

[1] Cochran, W. G. *Sampling Techniques*, 3rd edition, John Wiley and Sons, New York, NY, 1977, p. 89.

[2] Dalenius, T., and Hodges, J. L. (1959) Minimum Variance Stratification. *Journal of the American Statistical Association*, Vol. 54, pp. 88-101.

[3] Singh, R. (1971) Approximately Optimum Stratification on the Auxiliary Variable**.** *Journal of the American Statistical Association*, Vol. 66, pp. 829-833

[4] Lavallée, P., and Hidiroglou, M. A. (1988) On the stratification of skewed populations. *Survey Methodology*, Vol. 14, pp. 33-43.

[5] Sweet, E. M. and Sigman, R. S. Evaluation of Model-Assisted Procedures for Stratifying Skewed Populations Using Auxiliary Data. In *Proceedings of the Section on Survey Research Methods, 1995*. American Statistical Association, 1995, pp. 491-496.

[6] Cochran, W. G. *op. cit.* pp. 97-98.

[7] 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

[8] 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.

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

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

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

[12] Cochran, W. G., *op. cit*., pp. 97-98.

[13] 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.

[14] Bethel, J. *op. cit.*

[15] Särndal, C.-E., Swensson, B., and Wretman, J. *Model Assisted Survey Sampling*. Springer Verlang, New York, NY, 1991, pp. 106-110.

[16] Bethel, J. *op. cit.*

[17] *ibid.*

[18] Zayatz, L. and Sigman, R. S. (1995) Multivariate Allocation of Stratified Samples Using Chromy's Algorithm," *Economic Statistical Methods Report Series ESM-9502*. U.S. Bureau of the Census.