J. Ind. Soc. Agril. Statist.
61(3), 2007 : 355-360

Application of Evolutionary Algorithms in Multivariate Stratification

M. Kozak and S. Singh[1]
Department of Biometry and Bioinformatics, Warsaw University of Life Sciences, Poland
(Received : November 2005)

SUMMARY

In this article the evolutionary algorithm is applied to stratify a multivariate population. The algorithm is illustrated for agricultural data originating from the Agricultural Census 2002, conducted by the Central Statistical Office of Poland. The results obtained are compared with the results of a classical non-linear optimization method. Finally, the usefulness of global optimization methods in multivariate stratification is discussed.

Key words: Evolutionary algorithms, Multivariate stratification, Agricultural surveys.

1. INTRODUCTION

Lednicki and Wieczorkowski (2003) used the simplex method of Nelder and Mead (1965) to perform multivariate stratification. The simplex method is a local optimization method and hence may be inefficient. Kozak (2004) investigated this problem in a univariate stratification, and found that the random search method appeared to be more efficient than the simplex method. Niemiro (1999) applied the random search method in other problems of univariate stratification, whereas Keskinturk and Er (2007) proposed a genetic algorithm for univariate stratification. Therefore, there is a scope of improvement in multivariate stratification by making use of global optimization methods.

The aim of the paper is to show that evolutionary algorithms can be efficiently applied in multivariate stratification. The application of the algorithm is presented on the basis of a two-dimensional agricultural population and its results are compared with the results of the simplex method of Nelder and Mead (1965).

2. EVOLUTIONARY ALGORITHM FOR STRATIFICATION

In this section, the use of evolutionary algorithm for stratification is explained. Evolutionary algorithms are not elaborated here, for details one may refer to Goldberg (1989), Michalewicz (1992), Back (1996), Heitkoetter and Beasley (2001), and many others. The algorithm proposed in this paper is, in a way, a synthesis of the Genetic Algorithms (GAs), Evolutionary Programming (EP), and Evolution Strategy (ES).

Let us formulate the optimization problem first. Assume that a population U is subdivided into J subpopulations. Each j^{th}, j = 1, ..., J, subpopulation is to be stratified into L strata (where L is assumed fixed) using k auxiliary (stratification) variables. Let us assume that survey variables are strongly correlated with auxiliary variables, a common assumption in survey practice. We aim at finding such an array a of strata boundaries which minimises an overall sample size, i.e., the sum of sample sizes from the subpopulations, with respect to fixed coefficients of variation c_i (i = 1, ..., k, where k is the number of stratification variables) of estimators of the population mean of the variables in the subpopulations. The objective function is of the form

$$f(a_1,...,a_J) = \sum_{j=1}^{J} f(a_j) \qquad (1)$$

Given a_j (which are, in fact, the parameters we are looking for), the procedure of evaluation of the values

[1] *University of Texas at Brownsville and Texas Southmost College, Brownsville, Texas 78520, USA*

$f(a_j)$ in the j^{th} subpopulation is as follows (Lednicki and Wieczorkowski 2003). First, evaluate sample sizes for each Y_i, i = 1, ..., k

$$f_i(a_j) = N_{jL} + \left(\sum_{h=1}^{L-1} W_{jh} S_{jih} \right)^2 \left(\bar{Y}_{ji}^2 c_i^2 + N_j^{-1} \sum_{h=1}^{L-1} W_{jh} S_{jih}^2 \right)^{-1}$$

$$j = 1,... J \qquad (2)$$

where $f_i(a_j)$ is the minimum sample size for the i^{th} variable from the j^{th} subpopulation required to obtain the desired level of precision of estimation, S_{jih}^2 is the variance of the i^{th} variable in the h^{th} stratum of the j^{th} subpopulation, $\bar{Y}_{ji}$ is the population mean of Y_i in the j^{th} subpopulation, c_i is the fixed value of coefficient of variation (cv) of the estimator of the population mean of the i^{th} variable, N_j is the size of the j^{th} subpopulation, and $W_{jh} = N_{jh} / N_j$.

Next, on the basis of the values $f_i(a_j)$ evaluate a final objective function given by (Lednicki and Wieczorkowski 2003)

$$f(a_j) = N_{jL} + \sum_{h=1}^{L-1} n_{jh} \qquad (3)$$

where

$$n_{jh} = \max_{i=1,...,k} \left\{ \left(n_j - N_{jL} \right) a_{jih} \right\}, h = 1,..., L$$

$$n_j = \max_{i=1,...,k} \left\{ f_i \left(a_j \right) \right\}$$

$$a_{jih} = W_{jh} S_{jih} \left(\sum_{h=1}^{L-1} W_{jh} S_{jih} \right)^{-1}$$

Function (3) assumes the Neyman optimal sample allocation between strata, under the take-all stratum approach (Lednicki and Wieczorkowski 2003). The optimization problem is to find an array a of strata boundaries which minimizes the objective function (3) under constraints

$$N_{jh} \geq 2, \quad 2 \leq n_{jh} \leq N_{jh}, h = 1,..., L - 1 \qquad (4)$$

We consider here a situation in which distribution of the survey (and stratification) variables is right skewed. In such a situation an efficient procedure is to create a so called "take-all" stratum, in which all units are taken to the sample (Lavalee and Hidiroglou 1988, Rivest 2002, Lednicki and Wieczorkowski 2003). The take-all stratum approach is taken into account in the equations (1) – (4).

In such a stratification procedure, it is assumed that strata boundaries can be represented by the array a. Then a specific L-rot-180 stratification geometry is used; in the bi-variate case strata have a form of the capital L rotated through 180 degrees (Briggs and Duoba 2000, Lednicki and Wieczorkowski 2003), in which case stratum boundaries can be represented by an array a with elements

$$a = \begin{bmatrix} a_{11} & \cdots & a_{1k} \\ \vdots & \vdots & \vdots \\ a_{(L-1)1} & \cdots & a_{(L-1)k} \end{bmatrix} \qquad (5)$$

where a_{hi}, h = 1, ...; L – 1; i = 1, ..., k is the particular stratum boundary. On the basis of (5) we can indicate to which strata a particular unit belongs. Consider a unit stratum indicator function L(X, a) which for the m^{th} population element assumes a value

$$L_m(X_m, a) = \begin{cases} 1 & \text{if } X_{im} \leq a_{i1} \\ h & \text{if } a_{i(h-1)} < X_{im} \leq a_{ih} \text{ and } X_{i'm} \leq a_{hi'}, h = 2,...,H-1 \\ H & \text{if } X_{im} > a_{i(H-1)}, (i,i' = 1,...,k, i' \neq i,\ m = 1,...,N) \end{cases}$$

There are some software programs that enable the user to use genetic algorithms (e.g., a package rgenoud in the R language – see R Development Core Team 2005). Their algorithms are universal, i.e., they can work with any function. Certainly, adapting the evolutionary algorithm to the specificity of stratification may make the algorithm work faster and thus be more efficient.

Let us introduce some aspects of the algorithm, namely representation, initialization of a population, and operations of crossover and mutation.

Representation. The algorithm can work on arrays a which have been presented in (5). Then, formally a particular chromosome (the individual) would be represented by the array a, and each stratum boundary would be a particular allele of a chromosome. Next, the population would have a form of a block matrix $[a_I]$, where I, (I = 1, ..., m), is an index of the individual. But

an alternative approach is to use vector a′ consisting of strata boundaries, i.e., the rows of the array a, combined into a vector

$$a' = \left(a_{11},\ldots,a_{1(L-1)},\ldots,a_{k1},\ldots,a_{k(L-1)}\right)^T \qquad (6)$$

In such a case some operations, e.g., the crossover, are easier to carry out. Therefore, the vector a′ will be used as the chromosome (individual). Its elements will be called the alleles. However, the length of a′ must not be too small because some of the operations of the algorithm would not work. Let us assume that its length is not smaller than 6. In general, the proposed algorithm works most efficient when we stratify a population into more than two strata.

Initialization. Some individuals of the initial population can be generated by means of the univariate stratification method for each variable; for instance, approximate methods of Dalenius and Hodges (1959), Eckman (1959), or Mahalanobis (1952) can be used. In a case of random individuals, strata boundaries for the i^{th} auxiliary variable should be drawn from the interval $(\min(X_i), \max(X_i))$ and then sorted out by its values.

Crossover. Next two crossover operations, i.e., a one-point and two-point crossover, are introduced. An algorithm of the one-point crossover is as follows:

- Draw a random integer from an interval $\langle 2,(L-1)k-1\rangle$; it will be a point of exchanging the information, i.e., the crossover point.
- Copy all the information from a parent A from the start up to the crossover point, and copy all the information from a parent B from the crossover point to the end of the chromosome, i.e., the vector of strata boundaries a′. Create a second offspring in the same way, but "changing the sex of the parents", i.e., copy all the information from the parent B from the start up to the crossover point, and all the information from the parent A from the crossover point to the end of the chromosome. In such a way, the new chromosome gets the head of one parent's chromosome combined with the tail of the other (Heitkoetter and Beasley 2001). Sort each part of the chromosome regarding the variable X_i.

The two-point crossover algorithm is as follows:

- Draw two random different integers from an interval $\langle 2,(L-1)k-1\rangle$; they will be the crossover points.
- Copy the information from the parent A from the start up to the first crossover point, and from the second crossover point to the end of the parent A; copy the information from the parent B from the first crossover point to the second crossover point. Create a second offspring by "changing sex of the parents", like in the one-point crossover. Remember to sort the obtained offspring.

As one can see, the two-point crossover should not be used in case of a short vector a′. In each step of the algorithm we can choose the type of the crossover in a random way with a given probability.

Mutation. In GAs, a mutation probability p_m is often fixed as an inversion of the population size, i.e., $p_m = m^{-1}$, where m is a number of individuals in the population (De Jong 1975). In EP the mutation probability is much greater than in GAs. But considering our representation, such a small probability would lead to "impoverishing" of the population, and in consequence we could not obtain the optimal solution. Hence p_m should be quite large, for instance $p_m \geq 0.5$. The mutation procedure is as follows. For each allele of the mutating individual

- Generate a random number $u \sim U(0, 1)$
- If $u < p_m$, where p_m is the fixed mutation probability, go to the next step; else the allele does not change.
- Change the considered allele, i.e., the strata boundary for the i^{th} variable, by the rounded value of a variable $z \sim N(0, \sigma_z)$, where σ_z is fixed. (investigations detected that the efficient value of σ_z can be obtained from $\sigma_z = (10L)^{-1}\left[\max(X) - \min(X)\right]$).
- Finally sort the individuals in the same way as in the crossover procedure.

Selection. Selection ensures that the best individuals survive. The importance of selection is pointed out by Holland (1962a, b) in his elaborations on adaptive systems. An important aspect of the selection in the EAs is a fitness function. It characterizes the fitness of the individuals to the environment. In EAs, minimization of the objective function resolves itself into maximization of a fitness function.

There are many fitness functions. In our algorithm the following one was applied (Cheng and Gen 1994)

$$g(x_I) = \frac{\max_J f(x_J) - f(x_I) + \gamma}{\max_J f(x_J) - \min_J f(x_J) + \gamma}, I,J=1,...,m \qquad (7)$$

where $g(x_I)$ is the value of the fitness of individual x_I, $f(x_I)$ is the value of the objective function for the individual x_I, $\max_J f(x_J)$ is the maximum value of $f(x)$ in a particular generation, γ is the given positive coefficient (say $\gamma = 1$).

There are many types of selection, of which the most often used are the fitness proportional selection, roulette wheel, and stochastic tournament selection. We will use the last one. Its algorithm consists of the following steps

- Draw T individuals from the population (with equal probabilities); carry out the tournament - from among the T individuals, choose the one with the best fitness, i.e., with the greatest value of the fitness function; T is fixed, usually T = 3.
- Repeat the above procedure m – 1 times to get the population consisting of m – 1 new individuals.

Furthermore, combine the selection with the cloning of the best individual. This results in taking the best individual from the parental population to the new selected population, and then carrying out the selection for getting the rest of the m – 1 individuals.

Algorithm

A problem is to find a vector a′ (of length $(L-1)\times k$) which minimizes the objective function (1) under the constraints (4). Here is the algorithm

1. Fix the size of a population, i.e., a number m of individuals. Create the random population consisting of m individuals, i.e., m vectors a′.
2. For each individual check up the constraints (4). If they are not fulfilled, mark the objective function for the individual with some large value.
3. Carry out the crossover m times, to obtain 2m offsprings.
4. Carry out the univariate mutation for each offspring and each variable independently.
5. Clone the best individual and then carry out the stochastic tournament selection from the population consisting of 2m – 1 individuals (without the cloned one).
6. Repeat steps 2 – 5 a fixed number of times, say R. An individual with the best fitness in the last, R^{th}, generation is a solution of the problem.

One can change some elements of the algorithm body, e.g., we could decide to use some other form of the crossover, mutation, selection, or fitness function. We also have to decide about the population size (i.e., the number of individuals), number of iterations, and about a presence of determinant individuals in an initial population.

3. EXAMPLE

In this section, the algorithm is illustrated for an agricultural data set based on the Agricultural Census carried out by Central Statistical Office of Poland in 2002. Two stratification variables are cereals and potato area; the frame consists of the farms with an agricultural land larger than 2 hectares. The population is subdivided into 16 subpopulations (Voivodships, the main administrative provinces of Poland). The results of the algorithm are compared with the results of the simplex method of Nelder and Mead (1965).

The results obtained are presented in Table 3. Ten strata are constructed; the fixed coefficients of variations of the estimators of the population mean of the variables in the Voivodships are the same for both estimators: $c_1 = c_2 = c = 0.005$ (for each Voivodship).

Table 1. Sample sizes from provinces required to obtain a precision of mean estimation of two stratification variables (a cereals and potato area) equal $c_1 = c_2 = c = 0.005$, got by using the simplex method (S) and the proposed evolutionary algorithm

Province Code	S	AE	Province Code	S	AE
02	5,900	5,254	18	4,800	4,804
04	6,835	5,632	20	4,987	4,938
06	6,114	5,689	22	5,984	4,798
08	3,580	3,414	24	6,162	5,672
10	5,795	5,838	26	5,289	5,034
12	5,840	5,232	28	4,858	5,097
14	6,542	6,514	30	6,883	6,888
16	4,431	4,545	32	4,169	4,135

Source: Own calculations based on CSO of Poland data from the Agricultural Census 2002

Generally, the final results of the evolutionary algorithm are better than the results of the simplex method, i.e., the overall sample size is smaller. Note that the results of the simplex method are equally good or slightly better in five Voivodships (10, 16, 18, 28, and 30).

5. DISCUSSION AND CONCLUSIONS

The main aim of this paper is to show that global optimization methods can be efficient in multivariate stratification. One of the methods of global optimization, the random search method, was shown to be efficient in univariate stratification (e.g., Kozak 2004, Kozak and Verma 2006). In this paper we have shown that global optimization methods can be efficient in multivariate stratification as well. The evolutionary algorithm is shown to be more efficient than the simplex method of Nelder and Mead (1965). However, we do not claim that the proposed algorithm is really the best one. Based on our results, we just say that evolutionary algorithms should be considered as a method that can optimize the multivariate stratification.

In the example considered, 25 individuals (chromosomes) represented by vectors, and 50 generations are used. For some Voivodships the simplex method appeared to be little more efficient. This could be corrected by applying more generations in AE algorithm; certainly, it would result in a longer working of the algorithm. Eventually, the body of the algorithm could be changed to optimize its work.

We have presented application of the algorithm in a particular stratification problem, presented by Lednicki and Wieczorkowski (2003). However, the algorithm can also be applied for any other stratification problem, for instance, where we stratify a multivariate population subject to fixed sample size and aim at minimizing coefficients on variation of the estimators under study. In fact, in the proposed algorithm a form of the objective function and constraints does not matter. Just change the formulas in the algorithm, and it will work properly.

This discussion inclines towards the conclusion that evolutionary algorithms are worth recommending in multivariate stratification. The results and conclusions are preliminary in nature. Further studies should regard, first, the optimum form of the algorithm, and second, optimality of its parameters. However, the results and discussion from the papers by Kozak (2004) and Keskinturk and Er (2007) and from this paper show that in univariate and multivariate stratification one can efficiently make use of global optimization methods.

ACKNOWLEDGEMENT

This research was conducted when Marcin Kozak was the employee of the Central Statistical Office of Poland, and Sarjinder Singh was the employee of Department of Statistics, St. Cloud University, USA. The authors are very indebted to an English Editor Ms. Melissa Lindsay, Write Place Center, St. Cloud State University for her professional help.

REFERENCES

Back, T. (1996). *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York,

Briggs, J. and Duoba, V. (2000). STRAT2D: Optimal bivariate stratification system. *Statistics New Zealand* (http://www.stats.gov.nz).

Cheng, R. and Gen, M. (1994). Evolution program for resource constrained project scheduling problem. *Proc. of the 1st IEEE International Conference on Evolutionary Computation*, Florida, 736-741.

Dalenius, T. and Hodges, J.L. (1959). Minimum variance stratification. *J. Amer. Statist. Assoc.*, **54**, 88-101.

De Jong, K.A. (1975). An Analysis of the Behavior of a Class of Genetic Adaptive Systems. Doctoral dissertation, University of Michigan. *Dissertation Abstracts International*, **36(7)**, 5140B.

Eckman, G. (1959). An approximation useful in univariate stratification. *Ann. Math. Statist.*, **30,** 219-229.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Publishing Company, Inc.

Heitk ™ Oetter, J. and Beasley, D. (2001). The Hitch-Hiker's Guide to Evolutionary Computation. (http://www.cs.bham.ac.uk/Mirrors/ftp.de.uu.net/EC/clife/www).

Holland, J.H. (1962a). Information processing in adaptive systems. *Proceedings of the International Union of Psychological Sciences*, **3**, 330-339.

Holland, J.H. (1962b). Outline for a logical theory of adaptive systems. *J. Assoc. Compu. Machinery,* **3**, 297-314.

Keskintšrk T, ER (2007). A genetic algorithm approach to determine stratum boundaries and sample sizes of each stratum in stratified sampling. *Compu. Statist. Data Anal.*, **52**, 53-67.

Kozak, M. (2004). Optimal stratification using random search method in agricultural surveys. *Stat. Trans.*, **6(5)**, 797-806.

Kozak, M. and Verma, M.R. (2006). Geometric versus optimization approach to stratification: Comparison of efficiency. *Survey Methodology,* **32(2)**, 157-163.

Lavalle, P. and Hidiroglou, M. (1988). On the stratification of skewed populations. *Survey Methodology*, **14**, 3-43.

Lednicki, B. and Wieczorkowski, R. (2003). Optimal stratification and sample allocation between subpopulations and strata. *Stat. Trans.*, **6,** 287-306.

Mahalanobis, P.C. (1952). Some aspects of the design of sample surveys. *Sankhya*, 1-7.

Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs.* Springer-Verlag, Berlin.

Nelder, J.A. and Mead, R. (1965). A Simplex method for function minimization. *Compu. J.*, **7**, s. 308-313.

Niemiro, W. (1999). Optimal construction of strata using random search method. *Wiadomosci statystyczne,* **10**, 1-9.

R Development Core Team (2005). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria; URL http://www.R-project.org.

Rivest, L.P. (2002). A generalization of Lavallee and Hidiroglou algorithm for stratification in business surveys. *Techniques d'enquete*, **28**, s. 207-214. (http://www.mat.ulaval.ca/pages/lpr/).