A Genetic Algorithm Approach to the Selection of Near-Optimal Subsets from Large Sets

P. Whiting
Tillinghast,
71 High Holborn, London,
UK, WC1V 6TH
paul.whiting@towersperrin.com

P.W. Poon
Tillinghast,
71 High Holborn, London,
UK, WC1V 6TH
pui.wah.poon@towersperrin.com

J.N. Carter
Dept Earth Science and Engineering,
Imperial College, London,
UK, SW7 2AZ
j.n.carter@imperial.ac.uk

ABSTRACT

The problem attempted in this paper is to select a sample from a large set where the sample is required to have a particular average property. The problem can be expressed as an optimisation problem where one selects a subset of r objects from a group of n objects and the objective function is the mismatch between the required average property and that of a proposed sample. We test our method on a real-life problem which arises when we model the assets of a life insurance company in order to understand its risk, solvency and/or capital requirements.

In this paper we describe a genetic algorithm developed to solve the generic selection task. We demonstrate the algorithm successfully solving our test problem.

Categories and Subject Descriptors

I.2.8 [**Computing Methodologies**]: Artifical Intelligence—*Heuristic Methods*

General Terms

Algorithms

Keywords

Genetic Algorithm, Sampling, Selection, Economics

1. INTRODUCTION

None of the three standard genetic algorithms identified by Falkenauer[2] are easily applied to this problem although, like ordering and grouping problems, selection problems are NP-hard and so become more difficult to solve exactly when n is large. Real world manifestations might include: Selecting a representative sample of individuals from a customer database, who could then be included in a pilot programme or marketing exercise; Selecting shares to hold in a portfolio designed to track an index with low error; Selecting employees with a desired combination of skills, experience and availability to work on a particular project.

The generic structure of all of the problems described above is the same. There exists a population of objects, each characterised by a finite set of attributes. We can consider a particular set of attributes as being represented by a point in some high dimensional space. The statistics of this population is assumed to be known. We have available a sample, size n, of the population, which cannot be increased. Unfortunately, either the statistics of the sample are not a good approximation to those of the whole population, or the sample is too large for further analysis. What is required is a smaller subset whose statistics are close to those of the whole population.

GECCO'05, June 25–29, 2005, Washington, DC, USA.
ACM 1-59593-010-8/05/0006.

2. THE REAL WORLD PROBLEM

The problem we have studied arises when we model the assets of a life insurance company in order to understand its risk, solvency and/or capital requirements. We use a Monte Carlo simulation approach to make the assessment. We develop a computer model of the company and simulate what happens to the value of the company under a very large number of randomly generated asset scenarios. Each asset scenario projects returns for many different types of assets (stocks, bonds, etc) over a forty year period. By looking at the distribution of outcomes across these scenarios, we can understand the risk of ruin, solvency and capital requirements of the company[1].

3. SELECTION GENETIC ALGORITHM

3.1 Crossover

Given two 'parent' trial solutions, A and B, our crossover operator begins by re-ordering the genes (scenarios) in their respective strings so that every gene common to A and B appears on the left. Thus the two chromosomes each consist of a 'head' of common scenarios and a 'tail' of scenarios unique to the trial solution. We then perform the usual two-point cross-over on the tail sections. An example is shown in Figure 1.

3.2 Mutation

In keeping with generally accepted genetic algorithm methodology, we have used a mutation operator to introduce random mutations into 'offspring'. The mutation operator we have used replaces a small number of genes (scenarios) at random points in the string with scenarios randomly selected from those not already present in that trial solution. Mutation may occur in any part of the string (it is not restricted to the 'tail').

Parent trial solutions

A:(1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 ,10)

B:(9 , 7 , 5 , 3 ,15,11,12,13,14, 1)

Re-order parent strings so that common genes appear on the left

A:(1 , 3 , 5 , 7 , 9 : 2 , 4 , 6 , 8 ,10)

B:(1 , 3 , 5 , 7 , 9 :11,12,13,14,15)

2-point crossover on the tails

A:(1 , 3 , 5 , 7 , 9 : 2 |12,13,14| 10)

B:(1 , 3 , 5 , 7 , 9 :11| 4 , 6 , 8 |15)

Offspring trial solutions

A:(1 , 3 , 5 , 7 , 9 , 2 ,12,13,14,10)

B:(1 , 3 , 5 , 7 , 9 ,11, 4 , 6 , 8 ,15)

Figure 1: Crossover operator

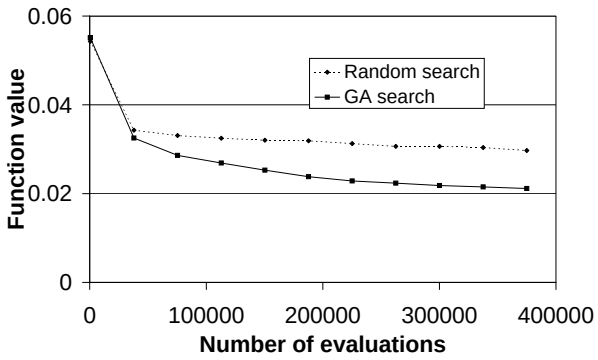

Figure 2: Comparison of average performance of random search and GA search

3.3 Other genetic algorithm parameters

The population size is 500 individuals, and the initial population is randomly selected from the set of all possible solutions using a uniform probability distribution function. At each generation 500 offspring are produced, and these completely replace the previous adult population (generational replacement scheme) with no elitism. In each generation 250 sets of two parents are selected randomly from the complete adult population of 500. The probability that an individual, i, is selected as a parent is given by

$$\frac{f_i}{\sum_j f_j} \text{ where } f_i = \frac{1}{1+v_i}$$

v_i is the objective function. The objective function will take on a value of zero for a subset which perfectly obtains the market consistency property. Each pair of parents produce two offspring using the crossover operator described above, 1% of the offspring are subject to mutation with five replacements taking place as described above. At each generation we keep track of the champion solution produced to date.

4. RESULTS

The test was to select a set of $r = 1000$ scenarios from a set of $n = 2000$ pre-selected scenarios. The preselection was such that the set of 2000 had approximately the same statistics as the whole population. Simple analysis had suggested that the set of 2000 scenarios was composed of many small groups that almost matched the statistics of the population. The task was therefore to identify these small groups and combine them so as to match the population statistics better. We considered using Simulated Annealing as a bench-mark algorithm. However, given this structure and the size of the local neighbour for a simple Simulated Annealing type algorithm, we would not expect Simulated Annealing to work well for the short duration tests that we wished to use. Our bench-mark algorithm was therefore a Random Search.

We ran the random search and the GA 20 times each. In figure 2 we can see the average performance of the two algorithms. Clearly the GA average is better that that of the random search, and the average function value of the champion is close to our target, of $v_i = 0.02$, after nearly 0.4 million function evaluations.

We conclude from these results that the performance of the GA is dependably better than the random search on this problem.

5. CONCLUSIONS

In this paper, we have shown how a genetic algorithm can be designed for a specific, real-life problem by applying some basic principles. We have demonstrated that resulting GA works on a large and complex real-life problem.

This work gives us confidence that this GA is a promising candidate for a generic genetic algorithm for selection tasks. However there are several things we might do which could be expected to improve its performance. These include: introduce an elitism strategy so as to increase the evolutionary pressure, reduce the population size so that it is easier to exploit the results of good solutions, introduce a tournament selection scheme. We also need to test alternative algorithms, even though our analysis of the problem suggest that standard algorithms, such as Simulated Annealing, are unlikely to work well.

6. ACKNOWLEDGEMENTS

The authors would like to thank the Tillinghast business of Towers Perrin, actuarial and management consultants to insurance and financial services companies, for supporting this research and for permitting the use of proprietary software for market consistent valuations and asset scenario generation.

7. REFERENCES

[1] True, S., and Rowland, J., Realistic Valuation: Dispelling the Myths, 2004 Life Convention, Institute of Actuaries, London, 2004.

[2] Falkenauer, E., Genetic Algorithms and Grouping Problems, John Wiley and Son, 1998.