Proc. of the 1991 IEEE
Int. Conf. on Tools for AI
San Jose, CA-Nov. 1991

Reducing The Search Time Of A Steady State Genetic Algorithm Using the Immigration Operator

Michael C. Moed — United Parcel Service, Research and Development, Danbury, CT

Charles V. Stewart — Dept of Comp. Sci., RPI, Troy, NY

Robert B. Kelly — ECSE Dept., RPI, Troy, NY

Abstract

This paper examines the fundamental trade-off between exploration and exploitation in a Genetic Algorithm (GA). An immigration operator is introduced that infuses random members into successive GA populations. It is theorized that immigration maintains much of the exploitation of the GA while increasing exploration. To test this theory, a set of functions are designed that often require the GA to perform an excessive number of evaluations to find the global optimum of the function. For these functions, it is shown experimentally that a GA enhanced with immigration (1) reduces the number of trials that require an excessive number of evaluations and (2) decreases the average number of evaluations needed to find the function optimum..

1 Introduction

The trade-off between exploration and exploitation in serial Genetic Algorithms (GA's) for function optimization is a fundamental issue [1]. If a GA is biased towards exploitation, highly fit members are repeatedly selected for recombination. Although this quickly promotes better members, the population can prematurely converge to a local optimum of the function. On the other hand, if a GA is biased towards exploration, large numbers of schemata are sampled which tends to inhibit premature convergence. Unfortunately, excessive exploration results in a large number of function evaluations, and defaults to random search in the worst case. To search effectively and efficiently, a GA must maintain a balance between these two opposing forces.

This study experimentally examines an immigration operator which, for certain types of functions, allows increased exploration while maintaining nearly the same level of exploitation for the given population size. To compare the performance of the GA with and without immigration, a suite of test functions is developed. Each function is characterized by different types of local and global optima. The local optima are designed to provide traps that the genetic algorithm must successfully avoid or recover from in order to achieve a global optimum. It is shown that a GA modified with immigration reduces the average number of evaluations required to find the function optimum over a range of population sizes. It is further shown that the number of trials requiring an excessive number of evaluations is reduced for functions in this set. Detailed results are not included in this paper, but are presented in [2].

2 A GA with the Immigration Operator

Population size is one parameter that directly effects the balance between exploration and exploitation. DeJong [3] noted that increasing the population size improves long-term performance of a GA at the expense of on-line performance for his test suite. In an extensive study, Schaffer et. al. [4] tested GA parameters for an expanded suite of functions and measured on-line performance. The study indicated that functions with many local optima have good on-line performance with larger population sizes. However, the study noted that an excessively large population imposes an increased number of evaluations per generation and produces poor overall performance.

To balance exploration with exploitation, we propose the following steady state algorithm that incorporates the immigration operator into the general structure of a Genetic Algorithm. A preliminary version of this algorithm was originally presented in [5]. Each iteration of the steady state algorithm we used is described as follows:

1. Evaluate each new member of the population

0-8186-2300-4/91 $01.00 © 1991 IEEE

and assign a fitness value.

2. Probabilistically select two members from the population based on fitness. These members are parents.

3. Perform one-point crossover on the parents at a random string position to form two children.

4. Probabilistically perform mutation on the children.

5. Replace the two worst members of the population with the children.

6. Probabilistically perform mutation on the rest of the population (optional).

The algorithm is modified to include immigration by the addition of the following step:

1.5. Generate and evaluate m random members and replace the m worst members of the population with the m random members.

With each generation of the GA, random members are immigrated and replace the worst members in the population. It is important to note that the number of random individuals substituted into the population each generation (m, called the immigration rate) is small compared to the size of the population. The advantages and trade-offs of immigration are described in [2].

3 Test Suite of Functions and Experiments

The experimental suite consists of a set of six functions that have different types of local and global optima. The suite was created to examine the ability of a GA to escape or avoid difficult local optima. This is reflected in the number of function evaluations required to find the global optimum of the function. By using a suite of this nature, one can determine the type of problems that prove difficult for a GA to solve, and show how the immigration operator affects performance. A description of the suite is presented in [2].

For the steady state GA, the increased exploration should prevent the GA from prematurely converging and becoming trapped in a local minima. This fact should be reflected in the number of GA trials that require an excessively long time to find the function optimum. Also, since the population size is maintained, the GA should have the exploration power of a larger population with the exploitation of a smaller one. This would be reflected in the average number of function evaluations required to find the global optimum. Therefore, for the steady state GA, two performance criteria are examined:

1. The average number of evaluations required to find the function optimum.

2. The number of trials that required an excessive number of evaluations to find the function optimum. These trials are called "outliers".

4 Experimental Results and Conclusion

The smallest average number of evaluations for five of the functions occurred when immigration was present. This was expected, since these functions contain many local optima. Further, the number of outliers for these functions was significantly reduced. For the sixth function, the steady state GA without immigration outperformed the modified GA. This occured because this function is unimodal, and does not require the increased exploration to efficiently find the function optimum.

It is interesting to note that for most of these functions, the reduction in function evaluations using immigration is largest for small populations, and decreases as population size increases. This indicates that the smaller populatoins required the added exploration that immigration provides, while larger populations possess sufficient exploration power.

In conclusion, this study demonstrates that immigration is a viable operator for improving the efficiency of GA's on difficult optimization problems.

References

[1] D. E. Goldberg, em Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, 1989.

[2] M. C. Moed, C. V. Stewart and R. B. Kelley, "Reducing the search time of a genetic algorithm using the immigration operator," submitted to *IEEE Trans. on Systems, Man, and Cybernetics*, 1990.

[3] K. A. De Jong, An Analysis of the Behavior of a Class of Genetic Adaptive Systems, PhD thesis, Univ. of Michigan, Ann Arbor, MI, 1975.

[4] J. D. Schaffer, R. A. Caruna, L. J. Eschelman, and R. Das, "A study of control parameters affecting on-line performance of genetic algorithms for function optimization," in *Proc. of the Third Int. Conf. on Genetic Algorithms*, pp. 51-60, 1989.

[5] M. C. Moed and G. N. Saridis, "A Boltzmann Machine for the organization of intelligent machines," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 20, no. 5, Sept., 1990.