Maintaining Population Diversity by Maintaining Family Structures

John Nicholson
North Carolina State University
Raleigh, North Carolina, USA
jwnichol@ncsu.edu

Mark White
North Carolina State University
Raleigh, North Carolina, USA
ncsu@markwhite.name

ABSTRACT

The selection event algorithm introduced an interesting method of creating offspring from parents in a multi-generational manner with a period of fitness neutrality followed by an intense selection event. This algorithm did not have a focus on maintaining diversity in the population, so it had all of the same pitfalls as other algorithms lacking such a focus. However, due to the novel multi-generational growth structure a natural family tree is created in the population, allowing for an equally natural diversity maintenance to be implemented which does not require any artificial diversity constraints to be placed on the fitness function. Instead, diversity is maintained (and encouraged) in the population through the growth dynamics of each family.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*

General Terms

Algorithms

Keywords

evolutionary strategy, diversity, selection

1. INTRODUCTION

A problem of primary concern in evolutionary algorithms is the loss of population diversity. When diversity is lost, the population is largely homogeneous and solution progress halts. While many algorithms have been developed with a focus on diversity, recently this focus has been increasing. Examples of mechanisms designed to maintain population diversity include fitness sharing [1], crowding [2], radius-based methods [7], clustering [8], island models or multi-populational GAs [4], and others. In addition, several efforts have been made to develop different measures of diversity [5].

The algorithm introduced in this paper is another multi-populational evolutionary strategy (ES). Since no crossover operations are performed migration across populations is moot, making the algorithm unique in that regard. Instead, the algorithm relies on small families randomly exploring the search space, then becoming large families when they find a good location to exploit. This is a similar motivation to Sewall Wright's Shifting Balance Theory [9].

2. SELECTION EVENT (SE) ALGORITHM

In [6], the selection event algorithm was introduced. In this algorithm, the population grows geometrically for consecutive generations before a periodic "selection event" occurs, causing the population size to return to its initial value. During the growth-phase, offspring are distributed uniformly to the parents (selection-free, see Figure 1). Then in the collapse phase (i.e., during the selection event), many tournaments are performed where the losers are eliminated from the population. This process tends to pick the best individuals in the population, regardless of their lineage. As a result, all-but-one of the initial seed individuals end up with no descendants after a short time.[1].

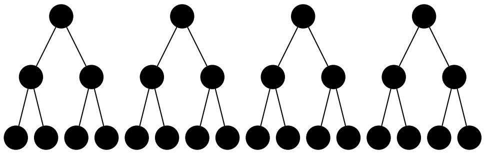

Figure 1: Family structure of SE algorithm: uniform growth

The SE algorithm can also be parameterized to be unable to uniformly distribute the offspring. In such a case, a similar illustration might look something like Figure 2.

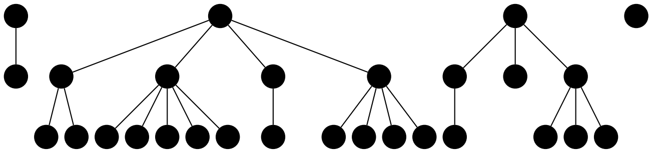

Figure 2: Family structure of SE algorithm: non-uniform growth

Non-uniform growth implies that there is competition between the families. The least-fit family may end up not having any offspring, killing the family and resulting in population diversity loss. Conversely, if one family is more-fit than other families, it will become larger because the members of that family will have more offspring than the members of

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-130-9/08/07.

[1] This effect is comparable to takeover[3]

the other families. This is a necessary condition for Sewall Wright's Shifting Balance Theory [9].

3. FAMILY-BASED SELECTION EVENT (FSE) ALGORITHM

The SE algorithm provides a mechanism for a new type of diversity maintenance. Due to the growth-phase of the algorithm, there is a natural *family tree* structure created, as illustrated in the figures. In these figures, there are four families – one for each initial individual – where all members of the same family share a common ancestry. If we were to change (1) the growth phase to prevent the death of a family, and change (2) the collapse phase of the SE algorithm to simply pick the best individual from each of the four families, the initial family structure would be maintained. It is the hypothesis of this paper that maintaining this family structure will help maintain (or encourage) diversity in the population, which will in turn help algorithm performance.

The FSE algorithm can be described in pseudo-code as shown in Algorithm 1. We break the production of new individuals into three components: guaranteed ($\mathcal{F}_I$ offspring per individual), through competition with other members of the same family ($\mathcal{F}_F$ offspring per individual, average), and through competition with all other individuals in the population ($\mathcal{F}_A$ offspring per individual, average). This compartmentalization allows for the control of how the population grows, including the ability to prevent the death of families. We also change the collapse-phase of the algorithm to be point number two above; that is, simply to pick the best individual in each family.

Algorithm 1 Family-based Selection Event (FSE) Algorithm

```
Initialize the population. The number of individuals is
the number of families.
while Not stopped do
  {Do the growth phase}
  for Each individual in the population do
    Create $\mathcal{F}_I$ offspring without competition.
    Create $\mathcal{F}_F$ offspring by competing with all other individuals in the family.
    Create $\mathcal{F}_A$ offspring by competing with all other individuals in the population.
  end for
  if Time for a periodic selection event then
    {Do the collapse phase}
    for Each family in the population do
      Keep the most-fit individual, and remove all others
    end for
  end if
end while
```

Please note:

1. The minimum *family* growth rate from one generation to the next is $\mathcal{F}_I + \mathcal{F}_F$.
2. In the first growth generation, before which each family has a size of one, there is always a single least-fit family who will grow at the minimum family growth rate.
3. Smaller family sizes allow for exploration of the problem (through random genetic drift if $\mathcal{F}_I + \mathcal{F}_F = 1$, otherwise through weak selection pressure), while larger family sizes are exploiting the problem and trying to find a local optimum.
4. If the family size is a constant one as a result of growth from only $\mathcal{F}_I$ or $\mathcal{F}_F$, that family is performing a random walk (completely selection free; all exploration).

4. EXPERIMENTS AND RESULTS

A simple trap test function was created to test the ability of the FSE algorithm to both have the ability for a family to escape from a local basin of attraction as well as find the global optimum. The algorithm has been shown to achieve both of these goals over a wide range of parameter values. This shows that even though there are many parameters introduced by this algorithm, the sensitivity to changes in their values on performance of the algorithm may not be that significant. By examining the dynamics of the population, we were also able to confirm the "shifting balance" motivation: small families were exploring the environment (with large mutations), while large families were exploiting the environment, refining the value of their local optimum with smaller and smaller mutations. Finally, we were able to show the population diversity in FSE generally increased, until the basin containing the global optimum was found.

Additionally, the FSE algorithm was tested on a common benchmark problem from the literature (the F8F2 function, a combination of Griewank's and Rosenbrock's functions [10], in six dimensions). We were able to show that the FSE algorithm is able to find the global optimum, as well as show that again the parameter values did not matter too much.

5. REFERENCES

[1] K. Deb and D. E. Goldberg. An investigation of niche and species formation in genetic function optimization. In *Proc. of the Third Int. Conf. on Genetic Algorithms*, pages 42–50, San Mateo, CA, 1989. Morgan Kaufmann.

[2] K. A. DeJong. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems.* PhD thesis, University of Michigan, Ann Arbor, MI, 1975. Dissertation Abstracts International 36(10), 5140B, University Microfilms Number 76-9381.

[3] D. E. Goldberg and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of Genetic Algorithms*, pages 69–93, San Mateo, 1991. Morgan Kaufmann.

[4] J. Grefenstette. Parallel adaptive algorithms for function optimization. Technical Report CS-81-19, Vanderbilt University, Nashville, TN, 1981.

[5] R. W. Morrison and K. A. D. Jong. Measurement of population diversity. *Lecture Notes in Computer Science*, 2310, 2002.

[6] J. Nicholson and M. White. Benefits of a periodic selection event in evolutionary strategy algorithms. In *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, pages 128–133, Vancouver, BC, Canada, 16-21 July 2006. IEEE Press.

[7] O. M. Shir and T. Bäck. Niching in Evolution Strategies. In *2005 Genetic and Evolutionary Computation Conference (GECCO'2005)*, volume 1, pages 915–916, New York, USA, June 2005. ACM Press.

[8] F. Streichert, G. Stein, H. Ulmer, and A. Zell. A Clustering Based Niching EA for Multimodal Search Spaces. In *Artificial Evolution, 6th International Conference, Evolution Artificielle, EA 2003, Revised Selected Papers*, pages 293–304, Marseille, France, Oct. 2004. Springer-Verlag. Lecture Notes in Computer Science Vol. 2936.

[9] S. Wright. The roles of mutation, inbreeding, crossbreeding and selection in evolution. In *Proceedings of the Sixth International Congress of Genetics*, volume 1, pages 356–366, 1932.

[10] D. Whitley, K. Mathias, S. Rana, and J. Dzubera. Building Better Test Functions. In *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 239–246, San Francisco, CA, 1995. Morgan Kaufmann.