Homogeneous Genetic Algorithms

Alexander Stanoyevitch
Mathematics Department
California State Univ.-Dominguez Hills
1000 Victoria Street, Carson CA 90747
(310) 243-3905

astanoyevitch@csudh.edu

ABSTRACT

In this note, we briefly describe a new type of genetic algorithm that is designed to mitigate one or both of the following two major difficulties that traditional genetic algorithms may suffer: 1. When the number of "active genes" needs to be held constant or kept within some prescribed range, and 2. When the set of genes is much larger than the set of active genes of feasible solutions under consideration. These homogeneous genetic algorithms use (unordered) sets to represent "active genes" in chromosomes rather than strings, and accordingly the selection, mating and mutation operators are (naturally) defined using set-theoretic operations. Homogeneous GAs have significantly outperformed traditional genetic algorithms for some typical problems in which these difficulties arise.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Search– *heuristic methods.* G.2.2 [**Graph Theory**]: *Network Problems.*

General Terms

Algorithms, performance

Keywords

Evolutionary computation, homogeneous genetic algorithms

1. ELEMENTS OF THE ALGORITHM

For a given discrete optimization problem, in a **homogeneous GA**, elements of the feasible solution set are represented by their corresponding (unordered) set of **active genes**. The solution space must first be represented by binary sequences in some predetermined way. For example, in a homogeneous GA, the chromosome that represents the following binary string:

0001000000100000000000000000000000000000000000000000000

would be {4,11} since only the 4th and 11th genes are active in this binary string.

The mating selection schemes of traditional GAs depend only on the fitness function, and so will carry over identically to homogeneous GAs. Homogeneous GAs, have the following natural (set theoretic) crossover and mutation operators that make these algorithms very amenable to the two difficulties mentioned above:

Crossover operator for mating: Once the mating pool is formed, parents are randomly paired off to mate using the following crossover scheme that we call *random mixing crossover*: Suppose that a mating pair of parents is represented by the following sets of active genes: $x = \{a_1, a_2, \cdots, a_n\}$ and $x' = \{a_1', a_2', \cdots, a_{n'}'\}$, where a_i, a_j' are the active genes. We let Int = $P \cap P'$ denote the set-theoretic intersection of the active genes of the parents, and we let s denote the number of elements of Int. If either parent coincides with this intersection (i.e., if either $n = s$ or $n' = s$), then the two offspring are identical with the parents (i.e., the parents are cloned into the next generation, but subject to the mutation operator). In all other cases, we randomly choose an integer $j \in \{1, 2, \cdots, \min(n, n') - s\}$, and then randomly choose two subsets of size j from the two parent chromosomes that lie outside of their common intersection: $z \subset x \,\square\, \text{Int}$ and $z' \subset x' \,\square\, \text{Int}$. (We are using the tilde $\square$ to denote relative complements of sets.) The two offspring are then defined by swapping the genes of these two subsets in the two parents. In set-theoretic notation, the offspring can thus be expressed as $y = (x \,\square\, z) \cup z'$, and $y' = (x' \,\square\, z') \cup z$.

Mutation operator: We employ a natural mutation scheme that we refer to as *random pool mutation*. The chromosomes of the new offspring (with the two that were cloned being excepted) are first randomly selected to undergo mutations with a specified probability p_{select}. For a selected chromosome x, a random number of genes j to be mutated is selected from the range $\{1, 2, \cdots, \min(\text{size}(x), \text{size}(A \,\square\, x))\}$. Here, A is simply the set of all available genes (available for all feasible solutions to the optimization problem) and we use "size(S)" to mean the size/cardinality of a set S. Then a random set of j genes is selected from x and is exchanged with a randomly selected set of j genes from $A \,\square\, x$.

Performance comparisons have been done on some important problems such as the *p*-center (facility location) problem, which is NP complete, as well as some other network logistics problems where it is important that the number of "active genes" remain fixed. For such problems, the homogeneous GAs have obtained significantly improved results over traditional GAs. We point out that in such comparisons, the parameters and time allowances of the GAs were made as similar as was possible (and indeed, the fitness functions that were employed were ones developed in the literature for traditional GAs) so as to make the comparisons fair.

GECCO'07, July 7–11, 2007, London, England, United Kingdom
ACM 978-1-59593-697-4/07/0007.