Genetic Algorithms in Multimodal Search Space

Marcos Gestal
University of A Coruña, Spain

Julián Dorado
University of A Coruña, Spain

INTRODUCTION

Genetic algorithms (GAs) (Holland, 1975; Goldberg, 1989) try to find the solution for a problem using an initial group of individuals—the population—where each one represents a potential solution.

Actually they are successfully applied in very different and actual fields (Yang, Shan, & Bui, 2008; Yu, Davis, Baydar, & Roy, 2008); nevertheless, GAs have some restrictions on a search space with more than a global solution or a unique global solution, together with multiple local optima. A classical GA faced with such a situation tends to focus the search on the surroundings of the global solution; however, it would be interesting to know a higher number of possible solutions for several reasons: precise information about the search space, easy implementation of the local solutions compared with the global one, simple interpretation of certain solutions compared with others, and so forth. To achieve that knowledge, an iterative process will be executed until reaching the desired goals. Such process will start with the grouping of the individuals into species that will independently search a solution in their environments; following, the crossover operation will involve individuals from different species in order not to leave unexplored any search space area. The process will be repeated according to the goals achieved.

BACKGROUND

Multimodal Problems

There are problems that do not exclusively have a unique global solution, but they have multiple optima, either global or local: the multimodal problems (Ehrgott, 2005).

For dealing with such type of problems, it is interesting to know the higher possible number of solutions. On one hand, the knowledge about the problem might not be complete; this fact leads to the uncertainty about the goodness of the obtained solution, as it cannot be guaranteed that no better solutions might be found at the unexplored search space. On the other hand, and even achieving the best solution, there might be other possible solutions that, due to different reasons (economy, simplicity), might be preferable.

BRIEF INTRODUCTION TO GENETIC ALGORITHMS

GAs are adaptive methods, generally used in problems of search and of parameter optimization, based on sexual reproduction and on the "survival of the fittest" theory (Tomassini, 1995; Beasley, Bull, & Martin, 1993; De Jong, 2002).

A population, a group of individuals where each one represents a potential solution that will evolve through different generations, is initially created. The best individual would tend to be kept after several evolutions, but other less-fitted individuals will be also kept in order to keep diversity. The diversity will enable that there might be individuals with different characteristics that could, some of them, be suitably adapted to the eventual changes on the environment.

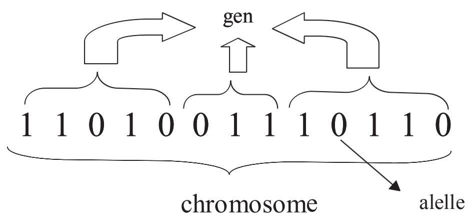
The best-fitted natural individuals are those that have more possibilities of having descendants, following the natural selection principles proposed of Darwin (1859).

In nature, individuals usually establish different groups, each of them specialized in different tasks: hunting, harvesting, and so forth. But the traditional GAs do not envisage this possibility for reaching several solutions; the present work will study an extension that will bear this in mind.

Problems Encoding

Any potential solution to a problem can be represented by providing values for a series of parameters. The whole of these parameters (or genes) is codified by a strand of values: the chromosome. The encoding is usually done by means of binary values, although other representations can also be used (see Figure 1).

Figure 1. Genetic individual



Main Algorithm

The generic functioning of a classic GA can be observed in Figure 2. A generation is obtained from a previous one by means of the reproduction operators. There are two types: the crossover and the copy. The crossover is a sexual reproduction that originates new descendants after the exchange of the genetic information of the parents. In the second case, a given number of individuals pass, with no variation, to the following population. Once the new individuals have been generated, the mutation is carried out with a P_m probability in such way that the transcription failures, which occurred in the copy of the genetic material during the sexual reproduction, are mimicked.

The GA run finishes when there are solutions good enough shaped as best individuals, when they all concur on a similar value, or when a prefixed generation's maximum number is reached.

For the GA to work correctly, a method should also exist that might indicate whether the population individuals represent or not, and to what extent, good solutions for the problem put forward. The later task would be carried out by the evaluation function, which establishes a numerical measurement (fitness) of the solution goodness (Koza, 1992).

EVOLUTIONARY APPROACHES TO MULTIMODAL PROBLEMS

Several approaches related to evolutionary techniques have been tried. A brief summary is shown in this section.

The modification proposed here is based on niching techniques. These are techniques that try to make and maintain stable subpopulations in GAs. This idea comes from nature, where individuals have different roles that allow them to survive in their natural ecosystems. These roles are called "ecological niches." Given some maximums, and given a limited capacity to locate them, the best niching algorithm will choose the global maximum. In addition, since they are not selective, they will keep both global and local maximums.

Different niching techniques have emerged throughout time. Some of the most important are:

- *Fitness Sharing:* This was first implemented by Goldberg and Richardson (1987) for its use with multimodal functions. This technique uses the concept of similarity

Figure 2: GA pseudocode

```
initialise current population randomly
WHILE the termination criterion is not fulfilled
 create temporal empty population
 WHILE temporal population don not fill
    select parents
    cross parents with Pc probability
   IF the crossover has occurred
               mutate one of the descendants with Pm probability
         assess descendants
        add descendants to the temporal population
   ELSE
      add parents to the temporal population
   END IF
END WHILE
increase generation counter
establish the temporal population as current new population
END WHILE
```