The methods of GA's modeling and their applications

Myung-Mook Han
(The Research Institute of Software Application, Yonsei University)

Abstract: *Genetic Algorithm(GA) is a parallel, global search technique modeled with the Darwinian principle of survival and reproduction of the fittest. Since Holland has proposed GA called the Simple GA, considerable research has focused on improving Simple GA performance. In this paper, I describe some methods of GA's modeling in different field.*

1 Introduction

Genetic Algorithms(GAs) are search techniques which were proposed by John Holland in 1975, and have been demonstrated to be robust in searching very large spaces in a wide variety of applications [1]. The search techniques may be classified into three classes: *Calculus-based techniques*, *Guided random search techniques*, and *Enumerative techniques*. *Calculus-based techniques* utilize a set of conditions which are satisfied by the solutions of an optimization problem. *Enumerative techniques* search one point at a time until every point associated with an specific domain space is examined. *Guided random search techniques*, which are further classed into simulated annealing and evolutionary computation, use not only enumerative methods but also additional information to conduct the search. Simulated annealing utilizes a thermodynamic process to converge to the global optimum. On the other hand, evolutionary computation is established on natural selection priniciples. These techniques subdivide into evolution strategies, evolutionary programming, and genetic algorithms. GAs are a repeated process which maintains a set of strings, called the population [2][3]. During each repeated step, the individuals in the current population are evaluated by the fitness function and as its base, the next population is formed through the application of genetic operators.

1.1 A brief history of Evolutionary Computation

In the 1950s and the 1960s, several computer scientists independently researched evolutionary systems which could be used as an optimization method. The idea in all these systems was to evolve a population of candidate solutions to a given problem, using operators imitated by natural genetic variation and natural selection. In the 1960s, Rechenberg introduced *evolutionary strategies*, a technique which optimizes real-valued parameters for devices. This technique was further developed by Schwefel. The field of evolution strategies has remained an area of active research. Fogel, Owens, and Walsh introduced *evolutionary programming*, a method that candidate solutions were represented as finite-state machines evolved by randomly mutating their state-transition diagrams and selecting the fittest. The field of evolutionary programming also remains an active area of research. In the 1960s, John Holland invented *genetic algorithms*, a method inspired the mechanisms of natural adaptation. His original goal was to study the phenomenon of adaptation as it occurs in nature and to develop the computer system. Holland's concept of a population-based algorithm with crossover and mutation was a major innovation.

1.2 Element of Genetic Algorithms

Most methods called GAs have at least four elements in common: populations of chromosomes, selection according to fitness, crossover to produce new offspring, and random mutation of new offspring.

populations of chromosomes The chromosomes in a population usually use the form of bit strings. Each locus in the chromosome has two possible alleles: 0 and 1. Each chromosome can be regarded as a point in the search space of candidate solutions. GAs perform populations of chromosomes, replacing one such population with another. GAs assign a fitness value of a fitness function to each chromosome in the current population. The fitness of a chromosome represents how well the chromosome solves the problem.

selection Selection operator chooses chromosomes in the population for reproduction. The more fit the chromosome, the more times it is likely to be chosen to reproduce.

crossover Crossover operator chooses a locus and exchanges the subsequences before and after that locus between two chromosomes to create two offspring.

mutation Mutation operator randomly flips some of the bits in a chromosome. The operator can occur at each bit position in a string with very small probability.

The fundamental concept of Simple GA is very simple, as illustrated in the following.

```
t = 0;
initialize population P(t);
fitness P(t);
While termination criterion
not reached
        t = t + 1;
        select P(t);
        crossover P(t);
        mutate P(t);
        fitness P(t);
```

Figure 1: The Simple Genetic Algorithm structure

1.3 Genetic Algorithms and traditional search methods

GAs are regarded as the search technique, but there are at least three(overlapping) meanings of search.

search for stored data Here the problem is to efficiently retrieve information stored in computer memory. Suppose you have a large database of names and telephone number stored in some ordered way. What is the best way to search for the record relating to a given last name? Typical algorithm is a binary search.

search for paths to goals Here the problem is to efficiently find a set of actions that will move from a given initial state to a given goal. This form of search is central to many approaches in artificial intelligence. The search algorithms considered in most AI field are methods for efficiently finding the best path in the tree from the initial state to the goal state. Typical algorithms are depth-first search, branch and bound, and A*.

search for solutions This is a more general class of search than the second class of search. The idea is to efficiently find a solution to a problem in a large space of candidate solutions. These are the kinds of search problems for which genetic algorithms are used.

Hill climbing, simulated annealing, and tabu search are other general methods for solving these problems. In AI, such general methods are called weak methods, and the specially designed methods to work on particular problems are called strong method.

2 Constrained Models

Recently there has been an increased interest in the application of Genetic Algorithm(GA) to the class of optimization problems [4]. The constrained optimization problems are quite numerous among optimization problems, and there has been many attempts to solve them using GAs. Attempts follows three different approach. The first way of dealing with candidates that violate the constraints is to generate potential solutions without considering the constraints and then to penalize them by decreasing the goodness of the evaluation function. In other words, a constrained problem is transformed to an unconstrained one by associating a penalty with all constraint violations; these penalties are included in the function evaluation. The second way of constraint handling methods is based on application of special repair algorithms to correct any infeasible solutions so generated. The third approach concentrates on the use of special representation mappings(decoders) which guarantee the generation of a feasible solution or the use of problem-specific operators which preserve feasibility of the solutions.

3 Hybrid Models

One of the most common forms of hybrid Genetic Algorithms is to incorporate local optimization as an add-on extra to the Simple GA loop of recombination and selection [5]. With the hybrid approach, local optimization is applied to each newly generated offspring to move it to a local optimum before injecting it into the population. GAs are used to perform global exploration among a population, while heuristic methods are used to perform local exploitation around chromosomes.

4 Parallel Models

GAs are good candidates for parallelization because they are modeled on a principle of evolving in parallel a population of individuals. However, there is no parallelization in the Simple GA which is implemented by the global control. Since it requires the serial execution and no-local interactions between the processors, Simple GA requires high communication costs. Several methods have been approached to overcome this limitation. These methods are group into two main models.

- the *distributed model* where several subpopulations perform in parallel, periodically exchanging their best individuals with the neighboring subpopulations.
- the *fine-grained model* where each individual is placed in a cell of a planar grid and genetic operators are applied between neighboring individuals on the grid.

These two models naturally are utilized on different parallel computer: MIMD machines, typically hypercube-based, for the distributed model, SIMD machines, array processors or connection machines for the fine-grained model.

4.1 The Distributed Model

There is the biological observation that isolated environments often produce species that are more adapted to their environments than areas of wider environments. It is known as the niching and speciation theory, which has inspired the GA researchers with new operators and structure for the population. Also, it is regarded that several competing subpopulations could search effectively.

The distributed model is based on a biological observation. That is, the population is divided into a number of isolated subpopulations, each of which performs independently and periodically swaps its worst individual(s) with the best individuals(s) of its neighbors. This swapping performance is called *migration* of individuals. The basic algorithm is as follows:

Step 1: Choose a genetic representation of the problem to solve. Generate randomly a population of individuals and partition it into n subpopulations.

Step 2: Each subpopulation perform Step 3 and 4.

Step 3: Apply the genetic operators to subpopulations.

Step 4: Replace n individuals of the population using the migration operator.

Step 5: If the algorithm is not finished, goto Step 2.

MIMD machines exhibit the best environment to use the distributed model fully. Several researchers have used MIMD machines approach to perform different GAs for different subpopulations.

Tanese first carried out research on the distributed model [6]. He implemented the distributed model on an hypercube computer, which is a general purpose 64 processor NCUBE/six hypercube. The experiments were performed for optimizing a Walsh Polynomial where each processor ran a subpopulation, periodically swapping good individuals with its neighbors.

This model has also been tested as various function optimizers. Research in this direction has been proposed by Mühlenbein, Schomisch and Born [7]. Parallel GA implemented a distributed model with a rank-based selection and a local hill-clinbing method. The experiments have been run on a multiprocessor MIMD machine which was designed in a ladder-like style.

4.2 The Fine-grained Model

The fine-grained model includes a concept of *spatiality*, in that the individuals of the population are assigned to a specific position in a two dimensional grid implemented as a toroidal array. Therefore, a neighborhood relation is determined by specifying spatial proximity among individuals and global control is not needed for this model. Selection is performed over the assigned individual and its neighboring individuals. The basic algorithm is presented below.

Step 1: Generate randomly a population of individuals and assign each of them to a cell of a grid. Calculate the fitness of each individual.

Step 2: For each current cell of the grid perform Steps 3, 4, 5 and 6.

Step 3: Choose the new individual to fill cell among those assigned to cell and to its neighbours.

Step 4: Randomly choose an individual in the neighbourhood of cell and crossover it with that assigned to cell. Assign one of the offspring to cell.

Step 5: Mutate the individual assigned to cell.

Step 6: Calculate the fitness of the new individual assigned to cell.

Step 7: If the algorithm is not finished, goto Step 2.

One of the first approaches dealing with this topic has been published by B.Manderick and P.Spiessens [8]. The paper describes in detail the fine-grained model and the performance of this model has been compared to that of a Simple GA, by means of online and offline performance measure.

J.Thomas Ngo and Joe Marks have also used this model to tackle a Spacetime Constraints problems which arises in the design of physically realistic animations that feature autonomous characters [9]. The implementation selects stimulus-response parameters using a parallel genetic algorithms which performs on a Thinking Machine CM-2 SIMD parallel machine. The processors are laid out on a 64*64 toroidal grid and each processor evaluates one individual. Also, they present the random-walk mate-selection scheme to select a mate.

5 Genetic-Based Machine Learning Models

Machine learning may be defined as any process which envolves designing computer programs to build new knowledge or to improve already possessed knowledge. One of the basic types of learning is inductive learning which learns any concepts or ideas by generalizing specific facts. It has received considerable attention in artificial intelligence. An important goal of inductive learning is to create a meaningful classification of observed objects or events.

Classification process is regarded as the first step in developing a theory about a collection of observations. This process is a form of learning from observation, and its purpose is to organize given observations into a hierarchy of meaningful categories.

Unitil now, the most of the past works on this field have been done under the title of numerical taxonomy. These methods are based on the application of a mathematical measure of similarity between objects. Each class of objects is taken as the collection of the objects whose intraclass similarity is high, and interclass similarity is low. However, such approaches have significant limitation because they do not take into consideration any background knowledge among the object attributes or the global concepts that could be used for characterizing the object configurations. As a result, classification obtained by the traditional method is often difficult to interpret conceptually.

To overcome this, conceptual clustering was introduced by Michalski and Stepp as an extension of processes of numerical taxonomy [10]. The main advantage of the conceptual clustering is the ability to capture the properties of object clusters that characterize a cluster as a whole and are not derivable from properties of individual entities.

CLUSTER/2 has been proposed as a conceptual clustering algorithm to make such classifications, but it requires NP-complete algorithms. Therefore, if the problem size becomes larger, the greater computaional complexity the more disadvantageous the method. Also, because CLUSTER/2 handles one concept at a time, the compuational cost is relevant. Finally, CLUSTER/2 introduces the *lexicographical evaluation functional*(LEF) to evaluate the quality of clustering. But there is no criteria for choosing one or another of the various modification. We propose the GAIL(Genetic Algorithm-based Inductive Learning) system which adapts GA as a key element and has the parallel characteristic. In GAIL, GA is presented for conceptual clustering that is based on a problem-specific representation method and problem-specific genetic operators. We will introduce the modified evaluation criteria to evaluate the quality of a clustering. We will use artificial data and implement two experiments which are evaluated by the modified evaluation criteria. As a result, we obtained the meaningful concepts of the events.

6 Conclusion

Genetic algorithms model concepts from the evolutionary process found in nature can be used to solve problems in a wide variety of domains. A population of candidate solutions is gradually improved by the genetic operators. Genetic algorithms are particularly appropriate for the complex optimization problems, and are good for applications which need adaptive problem solving strategies. When one wants to apply the GA to a particular problem, one faces a huge number of choices about how to proceed, with little theoretical guidance on how to make them. In this paper, I have described some modeling issues for GAs and some sophisticated GA techniques. Also I expect the number and diversity of application-oriented models to expand rapidly in the next several years.

References

[1] J.H.Holland, *Adaptation in Nature and Artificial Systems.* The Univ. of Michigan Press, Ann Arbor, MI, 1975.

[2] Lawrence Davis, ed., *Genetic Algorithms and Simulated Annealing.* Morgan Kaufmann Publishers, 1987.

[3] David E.Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison Wesley, 1989.

[4] Kenneth A.DeJong and William M.Spears, "Using Genetic Algorithms to Solve NP-Complete Problem," In *Proc. Third Int. Conf. on Genetic Algorithms,* pp.124-132, 1989.

[5] David E.Goldberg, *Computer-aided gas pipeline operation using genetic algorithms and rule learning,* Doctoral dissertation, Univ. of Michigan, 1983.

[6] Reiko Tanese, "Parallel Genetic Algorithm for a Hypercube," In *Proc. Second Int. Conf. on Genetic Algorithms,* pp.177-183, 1987.

[7] Mühlenbein, Schomisch and J.Born, "The Parallel Genetic Algorithm as Function Optimizer," In *Proc. Fourth Int. Conf. on Genetic Algorithms,* pp.271-278, 1991.

[8] B.Manderick and P.Spiessens, "The Parallel Genetic Algorithm as Function Optimizer," In *Proc. Third Int. Conf. on Genetic Algorithms,* pp.428-433, 1989.

[9] J.Thomas Ngo and Joe Marks, "Physically Realistic Motion Sysnthesis in Animation," In *Evolutionary Comp.,* vol.1, no. 3, pp.235-268, 1991.

[10] Michalski,R.S. and Stepp,R.E, "Learning from observation: Conceptual clustering," In *Machine Learning: An Artificial Intelligence Approach,* Tioga, Palo Alto, CA. pp.331-363, 1983.