A Survey of the Variants of Genetic Algorithm

Bhawana Singh

Abstract- This paper defines four different kinds of the enhanced genetic algorithm procedures which includes hybrid genetic algorithm, interval genetic algorithm, hybrid interval genetic algorithm and parallel genetic algorithm. The hybrid genetic algorithm result into a better optimum solution than the traditional optimization algorithms and genetic algorithm. The interval genetic algorithm and hybrid interval genetic algorithm avoid calculating system slope in traditional interval analysis and determines the optimum interval range of the parameters under allowable corresponding objective error boundary. The parallel genetic algorithm develops very accurate schedules when the parameter guidelines are used.

Index Terms – Genetic Algorithm, Hybrid Genetic Algorithm, Hybrid Interval Genetic algorithm, Interval Genetic Algorithm, Parallel Genetic Algorithm.

1 INTRODUCTION

Optimization is the process of making something better. It is a process that finds a best, or optimal, solution for a problem. The Optimization problems are centered around three factors: an objective function which is to be maximized or minimized; a set of unknowns or variables that affect the objective function and a set of constraints that allows the unknown to take on certain values but exclude others.

Genetic Algorithms are a family of computational models inspired by evolution. These algorithms encode a potential solution to a specific problem on a simple chromosome like data structure and apply recombination operators to these structures so as to preserve critical information. Genetic algorithms are often viewed as function optimizers, although the range of problems to which genetic algorithms have been applied is quite broad.

An implementation of a genetic algorithm begins with a population of typically random chromosomes. One then evaluates these structures and allocates reproductive opportunities in such a way that those chromosomes which represent a better solution to the target problem are given more chances to reproduce than those chromosomes which are poorer solutions. The goodness of a solution is typically defined with respect to the current population.

There are three major advantages when applying the GA to optimization problems.

- It is better than conventional AI; It is more robust.
- GA's do not break easily even if the inputs changed slightly.
- While performing search in large state-space, genetic algorithms offer significant benefits over many other typical search optimization techniques.

- *Bhawana Singh is currently pursuing master's degree program in computer science & engineering in Invertis University, India. E-mail: bhawana1622.bs@gmail.com*

Even though the GA can find the solution in the whole problem domain, it fails to solve complexes constraint problems easily, especially for exact constraints. Also it is time-consuming when huge computations are required for populations and generations. To overcome the defects and employ the advantages of the GA, the enhanced GA is applied for the optimization design.

2 GENETIC ALGORITHM

Genetic Algorithm was developed by John Holland, University of Michigan (1970's). It is an adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetics. Genetic Algorithms are inspired by Darwin's theory about evolution-*"Survival of the fittest"*. It has been quite successfully, applied in machine learning and optimization problems. To solve a problem, a GA maintains a population of individuals (also called strings or chromosomes) and probabilistically modifies the population by some genetic operators such as selection, crossover and mutation, with the intent of seeking a near optimal solution to the problem [1].

A simple genetic algorithm has been shown below:

```
Simple_Genetic_Algorithm()
        {
        Initialize the Population;
        Calculate Fitness Function;

        While(Fitness Value != Optimal Value)
                {
                Selection; //Natural Selection, Survival Of Fittest
                Crossover; //Reproduction, Propagate favorable
                        characteristics
                Mutation;
                Calculate Fitness Function;
                }
        }
```

Fig 1. Pseudo code of Genetic algorithm

2.1 Encoding

The process of representing the solution in the form of a string that conveys the necessary information is called encoding. Just as in a chromosome, each gene controls a particular characteristic of the individual, similarly, each bit in the string represents a characteristic of the solution. There are various encoding techniques. One of them is the binary encoding. It is the most common method of encoding. Chromosomes are strings of 1s and 0s and each position in the chromosome represents a particular characteristic of the problem.

Chromosome A	10110010110011100101
Chromosome B	11111110000000011111

Fig 2. Representation of a chromosome

2.2 Fitness Function

A fitness function quantifies the optimality of a solution (chromosome) so that particular solution may be ranked against all the other solutions. A fitness value is assigned to each solution depending on how close it actually is to solving the problem. Ideal fitness function correlates closely to goal + quickly computable. For example, in TSP, f(x) is sum of distances between the cities in solution. The lesser the value, the fitter the solution is.

2.3 Recombination

The process that determines which solutions are to be preserved and allowed to reproduce and which ones deserve to die out is called recombination. The primary objective of the recombination operator is to emphasize the good solutions and eliminate the bad solutions in a population, while keeping the population size constant.

2.4 Crossover

It is the process in which two chromosomes (strings) combine their genetic material (bits) to produce a new offspring which possesses both their characteristics. Two strings are picked from the mating pool at random to cross over. The method chosen depends on the Encoding Method**.** The crossover operator starts with two selected individuals and then the crossover point, which is an integer between 1 and *N*-1, where *N* is the length of strings, is selected randomly.

Chromosome1	11011 | 00100110110
Chromosome 2	11011 | 11000011110
Offspring 1	11011 | 11000011110
Offspring 2	11011 | 00100110110

Fig 3. Representation of crossover operation

2.5 Mutation

It is the process by which a string is deliberately changed so as to maintain diversity in the population set. Mutation involves reordering of the list:

* *

Before: (5 8 7 2 1 6 3 4)

After: (5 8 6 2 1 7 3 4)

Just FLIP-BIT

3 HYBRID GENETIC ALGORITHM

It combines the Genetic Algorithm with the traditional optimization methods, in order to overcome the difficulty of a long-trial and error process that is faced in finding better initial values of the design variables. The GA is applied to provide a set of initial design variables, thereby avoiding the trial process; thereafter, traditional algorithms are employed to determine the optimum results. This algorithm is more effective than traditional algorithms.

Steps involved are as follows:

Step 1: Produce initial population in random, let it be (t), and initialize t=0.

Step 2: Encode the string into binary (or other forms based on the encoding methods).

Step 3: Calculate fitness of the individuals in the population.

Step 4: Select the best fit from current population.

Step 5: Perform crossover on the selected strings.

Step 6: Perform mutation.

Step 7: Increment t=t+1.

Step 8: Check if generation is complete or not. If no, then goto step 4.

Step 9: Initialize initial population t1 to Z.

Step 10: Perform optimization by using the traditional method with initial design variable Z.

Step 11: Check if converged or not, that is, a better solution is found or not. If yes then exit, else assign Z to Z0 and goto step 10.

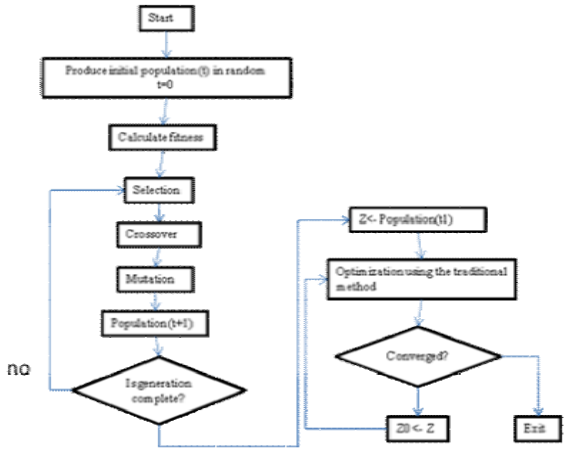

Fig 4. Flowchart of HGA

4 INTERVAL GENETIC ALGORITHM

It usually is not easy to manufacture exact design variables in engineering, because of measurement inaccuracies or errors within the manufacturing process itself. Furthermore, to exact manufacturing is often more costly. Interval optimization is one of the algorithms that can overcome these difficulties. The interval genetic algorithm seems faster and more reliable than the well-known simulated annealing, since the search for the global minimum is done using a population of points. Because of its complexity the proposed method is well suited for the optimization of multi modal functions; in simpler cases a local minimization procedure. The interval genetic algorithm can be made parallel with a speed-up close to the population size.

Steps of the interval genetic algorithm are as follows:

Step 1: Set the values of control parameters in the algorithm. Let t = 0, and initialize a population denoted as x(t).

Step 2: An interval fitness is assigned on an individual.

Step 3: Produce parents by selection.

Step 4: Perform crossover and mutation operators according to Formulas, and produce off-spring. Let t = t + 1.

Step 5: Judge if the termination criterion is met, if yes,

Step 6: then go to step 7; otherwise go to step 2.
Step 7: Output the optima and stop the algorithm.

Formulas applied are as follows:

The probability of crossover p_c (x_i (t)) of an individual x_i (t) is given as follows:

$$p_c(x_i(t))=1/(1+\exp(-k_1.(T/t).((\delta_1(t)+\delta_2(t))/(m(f_s(t))-m(f(x_i(t))))))$$

where, T is the termination generation, and k_1 is an adjustable coefficient. $\delta_1(t)$ and $\delta_2(t)$ gives the change of individual's fitness during a population evolution and $F_s(t)$ is called a superior fitness interval.

The probability of mutation operator p_m (x_i (t)) of an individual x_i (t) is as follows:

$$p_m(x_i(t))=1/(1+\exp(k_2.(T/t).((\delta_1(t)+\delta_2(t))/(m(f(x_i(t)))-m(f_i(t))))$$

where, k_2 is another adjustable coefficient.

5 HYBRID INTERVAL GENETIC ALGORITHM

In addition to the IGA, the Hybrid Interval Genetic Algorithm (HIGA) even combines the IGA with the Finite Element Method (FEM). This hybrid algorithm can exclude equations formulation and interval analysis, and determines the optimum interval parameters.

6 PARALLEL GENETIC ALGORITHM

Due to increasing demands placed on genetic algorithms, such as searching large search spaces with costly evaluation functions and using large population sizes, there is an ever growing need for fast implementations to allow quick and flexible experimentation. Parallel processing is the natural route to explore. Furthermore, some of the difficulties that face standard GAs (such as premature convergence) may be less of a problem for parallel variants.

Parallel genetic algorithms (PGA) use two major modifications compared to the genetic algorithm. Firstly, selection for mating is distributed. Individuals live in a 2D world. Selection of a mate is done by each individual independently in its neighborhood. Secondly, each individual may improve its fitness during its lifetime by e.g. local hill-climbing. The PGA is totally asynchronous, running with maximal efficiency on MIMD parallel computers. The search strategy of the PGA is based on a small number of intelligent and active individuals, whereas a GA uses a large population of passive individuals.

The basic idea behind most parallel programs is to divide a task into chunks and to solve the chunks simultaneously using multiple processors. This divide-and-conquer approach can be applied to GAs in many different ways. Some parallelization methods use a single population, while others divide the population into several relatively isolated subpopulations. Some methods can exploit massively parallel computer architectures, while others are better suited to multicomputer with fewer and more powerful processing elements.

The way in which GAs can be parallelized depends on the following elements:
• How fitness is evaluated and mutation is applied
• If single or multiple subpopulations (demes) are used

• If multiple populations are used, how individuals are exchanged
• How selection is applied (globally or locally)

There are three main types of parallel GAs:
(1) Global single-population master-slave GAs,
(2) single-population fine-grained, and
(3) multiple-population coarse-grained GAs.

In a master-slave GA there is a single population (just as in a simple GA), but the evaluation of fitness is distributed among several processors. Since in this type of parallel GA, selection and crossover consider the entire population it is also known as global parallel GAs.

Fine-grained parallel GAs are suited for massively parallel computers and consist of one spatially-structured population. Selection and mating are restricted to a small neighborhood, but neighborhoods overlap permitting some interaction among all the individuals.

Multiple-population (or multiple-deme) GAs are more sophisticated, as they consist of several subpopulations which exchange individuals occasionally. This exchange of individuals is called migration and it is controlled by several parameters. Multiple-deme parallel GAs are known with different names. Sometimes they are known as "distributed" GAs, because they are usually implemented on distributed memory MIMD computers. Since the computation to communication ratio is usually high, they are occasionally called coarse-grained GAs. Finally, multiple-demes GAs resemble the "island model" in Population Genetics which considers relatively isolated demes, so the parallel GAs are also known as "island" parallel GAs.

The final method to parallelize GAs combines multiple demes with master-slave or fine-grained GAs. This class of algorithms is known as hierarchical parallel GAs, because at a higher level they are multiple-deme algorithms with single-population parallel GAs (either master-slave or fine-grained) at the lower level. A hierarchical parallel GAs combines the benefits of its components, and it promises better performance than any of them alone.

7 CONCLUSION

Many researches have been done on these enhanced genetic algorithms. Hybrid genetic algorithm and parallel genetic algorithm yield efficient result than simple genetic algorithm. Hybrid genetic algorithm is more reliable and faster than interval genetic algorithm. Parallel genetic algorithms show a high performance for solving the problems with multi objective functions. Classifications of parallel genetic algorithm can be used for various optimization problems.

REFERENCES

[1] P. Guo, X. Wang, Y. Han, "The Enhanced Genetic Algorithms for the Optimization Design", School of Civil and Architectural Engineering, Liaoning University of Technology, Jinzhou, China, 2010 3rd International Conference on Biomedical Engineering and Informatics (BMEI 2010).

[2]D. G. Sotiropoulos, E. C.Stavropoulos, M. N. Vrahatis, "A New Hybrid Genetic Algorithm For Global Optimization", Department of Mathematics, University of Patras, Greece, Nonlinear Analysis Theory, Methods & Applications, 30 (7), pp. 4529-4538,1997.

[3]Darrell Whitley, Doug Hains, Adele Howe, "A Hybrid Genetic Algorithm for the Traveling Salesman Problem using Generalized Partition Crossover ", Colorado State University, Fort Collins, CO 80524.

[4] Ardhendu mandal, Jayanta Dutta and S.C.Pal, "A new efficient technique to construct a minimum spanning tree", University of North Bengal; Volume 2, Issue 10, October 2012 ISSN: 2277 128X ;International Journal of Advanced Research in Computer Science and Software Engineering.

[5] Macro Muselli and Sandro Ridella, "Global Optimization Of Functions With The Interval Genetic *Algorithm*".

[6] K.Zmanifar and M.Koorangi, "Designing optimal binary search tree using parallel genetic algorithms", Department of Computer, Faculty of Engineering, University of Isfahan, Iran.

[7] Melanie Mitchell and Stephanie Forrest, "Genetic Algorithms and Artificial Life", In Artificial Life, 1 (3), 267-289.

[8] Marko, Marin Golub and Leo Budin,"Solving n-Queen problem using global parallel genetic algorithm", EUROCON 2003 Ljubljana, Slovenia.