Volume 5, Issue 7, July 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Review: Comparison of Genetic Algorithms and Particle Swarm Optimization

Anu Taneja
Assistant Professor, BCIIT, IP University,
Delhi, India

Abstract: ***Nature-Inspired Algorithms have been gaining much popularity in recent years due to the fact that many real-world optimization problems have become increasingly large, complex and dynamic. The size and complexity of the problems nowadays require the development of methods and solutions whose efficiency is measured by their ability to find acceptable results within a reasonable amount of time, rather than an ability to guarantee the optimal solution. Thus 'Nature-Inspired Algorithms for Optimization' is a collection of the latest state-of-the-art algorithms and important studies for tackling various kinds of optimization problems.***

Keywords: GA-Genetic Algorithms, PSO-Particle Swarm Optimization.

I. INTRODUCTION OF GENETIC ALGORITHMS

The *genetic algorithm* (GA), developed by John Holland and his collaborators in the 1960s, is a model or abstraction of biological evolution based on Charles Darwin's theory of natural selection. Holland was probably the first to use the crossover and recombination, mutation, and selection in the study of adaptive and artificial systems. These genetic operators form the essential part of the genetic algorithm as a problem-solving strategy. Since then, many variants of genetic algorithms have been developed and applied to a wide range of optimization problems, from graph coloring to pattern recognition, from discrete systems (such as the traveling salesman problem) to continuous systems (e.g., the efficient design of airfoil in aerospace engineering), and from financial markets to multi-objective engineering optimization.

There are many advantages of genetic algorithms over traditional optimization algorithms. Two of the most notable are: the ability to deal with complex problems and parallelism. Genetic algorithms can deal with various types of optimization, whether the objective (fitness) function is stationary or non-stationary (changes with time), linear or nonlinear, continuous or discontinuous, or with random noise. Because multiple offsprings in a population act like independent agents, the population (or any subgroup) can explore the search space in many directions simultaneously. This feature makes it ideal to parallelize the algorithms for implementation. Different parameters and even different groups of encoded strings can be manipulated at the same time.

However, genetic algorithms [1] also have some disadvantages. The formulation of a fitness function, the use of population size, the choice of important parameters such as the rate of mutation and crossover, and the selection criteria of the new population should be carried out carefully. Any inappropriate choice will make it difficult for the algorithm to converge or it will simply produce meaningless results. Despite these drawbacks, genetic algorithms remain one of the most widely used optimization algorithms in modern nonlinear optimization.

II. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) was developed by Kennedy and Eberhart in 1995 based on swarm behavior in nature, such as fish and bird schooling. Since then, PSO has generated much wider interests and forms an exciting, ever-expanding research subject, called swarm intelligence. PSO has been applied to almost every area in optimization, computational intelligence, and design applications.

The PSO algorithm searches the space of an objective function by adjusting the trajectories of individual agents, called *particles*, as the piecewise paths formed by positional vectors in a quasi-stochastic manner. The movement of a swarming particle consists of two major components: a stochastic component and a deterministic component. Each particle is attracted toward the position of the current global best $\boldsymbol{g}*$ and its own best location $\boldsymbol{x}*\ i$ in history, while at the same time it has a tendency to move randomly. When a particle finds a location that is better than any previously found locations, updates that location as the new current best for particle i . There is a current best for all n particles at any time t during iterations. The aim is to find the global best among all the current best solutions until the objective no longer improves or after a certain number of iterations.

III. COMPARISON OF GENETIC ALGORITHMS AND PARTICLE SWARM OPTIMIZATION

PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA,

PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles.
Compared to GA, the advantages of PSO are that PSO [3] is easy to implement and there are few parameters to adjust. PSO has been successfully applied in many areas: function optimization, artificial neural network training, fuzzy system control, and other areas where GA can be applied.
PSO Procedure: PSO [1] is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In each iteration, every particle is updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called gbest. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called lbest.
After finding the two best values, the particle updates its velocity and positions with following equation (a) and (b).
v[] = v[] + c1 * rand() * (pbest[] - present[]) + c2 * rand() * (gbest[] - present[]) ---(a)
present [] = present[] + v[] ---(b)
v [] is the particle velocity, present[] is the current particle (solution).
pbest[] and gbest[] are defined as stated before.
rand () is a random number between (0,1).
c1, c2 are learning factors. usually c1 = c2 = 2.
The pseudo code of the procedure is as follows:

```
For each particle
    Initialize particle
END
Do
    For each particle
        Calculate fitness value
        If the fitness value is better than the best fitness value (pBest) in history then
            Set current value as the new pBest
   End
    Choose the particle with the best fitness value of all the particles as the gBest
    For each particle
        Calculate particle velocity according equation (a)
        Update particle position according equation (b)
   End
While maximum iterations or minimum error criteria is not attained.
```

Genetic Algorithms Procedure: Most of evolutionary techniques have the following procedure:

1. Random generation of an initial population
2. Reckoning of a fitness value for each subject. It will directly depend on the distance to the optimum.
3. Reproduction of the population based on fitness values.
4. If requirements are met, then stop Otherwise go back to 2.

From the procedure, we can learn that PSO shares many common points with GA. Both algorithms start with a group of a randomly generated population, both have fitness values to evaluate the population. Both update the population and search for the optimum with random techniques. Both systems do not guarantee success. However, PSO [2] does not have genetic operators like crossover and mutation. Particles update themselves with the internal velocity. They also have memory, which is important to the algorithm.
Compared with genetic algorithms (GAs), the information sharing mechanism in PSO is significantly different. In GAs, chromosomes share information with each other. So the whole population moves like a one group towards an optimal area. In PSO, only gBest (or lBest) gives out the information to others. It is a one -way information sharing mechanism. The evolution only looks for the best solution. Compared with GA, all the particles tend to converge to the best solution quickly even in the local version in most cases.

IV. CONCLUSION

So these are optimization algorithms, which start from an initial guess. After a certain (sufficiently large) number of iterations, it may converge toward to a stable solution, ideally the optimal solution to a problem of interest. This is essentially a self-organizing system with solutions as states and the converged solutions as attractors. Such an iterative, self-organizing system can evolve according to a set of rules or mathematical equations. As a result, such a complex system can interact and self-organize into certain converged states, showing some emergent characteristics of self-organization.

REFERENCES

[1] Nature Inspired optimization Algorithms by Xin-She Yang.
[2] http://web.mit.edu/deweck/www/PDF_archive/3%20Refereed%20Conference/3_50_AIAA-2005-1897.pdf
[3] http://www.sciencedirect.com/science/article/pii/S1568494607001330