IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727
PP 18-22
www.iosrjournals.org

A Comparative Study of Genetic Algorithm and Particle Swarm Optimization

Mrs.D.Shona[1], Dr.M.Senthilkumar[2], S.Valarmathi[3] & G.Malathy[4]
[1]*Research Scholar, Bharathiar University, Assistant Professor, Department of Computer Science, Sri Krishna Arts and Science College, Coimbatore.*
[2]*Assistant Professor, PG & Research Department of Computer Science, Government Arts College, Udumalpet.*
[3,4]*M.Sc CS, Department of Computer Science, Sri Krishna Arts and Science College, Coimbatore.*

***Abstract*:** *Among different search and optimization techniques, the development of Evolutionary Algorithms (EA) has gained large attention in the field of research and EA acts as an effective method for solving various optimization problems. Genetic Algorithm (GA) is an adaptive metaheuristic optimization algorithm which is mainly based on the theory of Natural Selection and Genetics. GA can be mainly used in various areas such as Neural Networks, Image Processing, Vehicle routing problems and so on. The drawback of the GA is, it is of high cost. Particle Swarm Optimization (PSO) which is also a kind of metaheuristic optimization algorithm based on the social behavior of organisms which possess swarming characteristics. PSO and the GA are very closely associated population based search methods where both advances from a set of points to another set of points in an iteration providing perceptible improvements from the antecedent values using some probabilistic and deterministic conditions. The main objective of this paper is to study and compare the efficiency and effectiveness of the two evolutionary algorithms.*

***Keywords*:** *Evolutionary Algorithm (EA), Genetic Algorithm (GA), Metaheuristics, Particle Swarm Optimization (PSO).*

I. Introduction

The Genetic Algorithm (GA) was introduced in the mid-1970s by John Holland and his colleagues and students at the University of Michigan [1]. Mostly GA provides very approximate solutions to various problems. GA uses various biological techniques such as inheritance, selection, crossover or recombination, mutation and reproduction. The GA uses the principle of "survival of the fittest" in its search process to choose and generate individuals which are adapted to the environment. The GA is applied to solve complex design optimization problems as it can handle both discrete and continuous variables and non-linear objective and constrain functions without requiring gradient information. Mainly, the optimization problems can be solved easily through the concept called Swarm Intelligence. The concept of Swarm Intelligence was introduced by Gerardo Beni and Jing Wang in 1989, who got inspired from bird flocking, ant colonies, fish schooling and animal herding. Particle Swarm Optimization (PSO) was developed by Kennedy and Eberhart in the mid-1990s [2]. The basic idea in PSO is that each particle represents a potential solution which it updates accordingly using decision process. There are no evolution operators such as crossover and mutation in PSO, rather it is provided with particles which fly through the problem space by following the current optimum particles. Due to its various advantages like efficiency, robustness, effectiveness and simplicity, PSO has been used large in numbers. When compared with other stochastic algorithms it has been found that PSO requires less computational effort [3][4]. Although PSO has shown its potential on many aspects for solving different optimization problems, it still requires considerable execution time to find solutions for large-scale engineering problems [5][6]. In section II, survey has been made on related works and in section III and IV, PSO versus GA and Comparison Measure are shown respectively. The Concluding remarks are given in section V.

II. Literature Survey

J. Eberhart et al [2] introduced the concept for the optimization of non-linear functions using particle swarm methodology. The evolution of several paradigms is outlined, and an implementation of one of the paradigms is discussed. Benchmark testing of the paradigm is described, and applications, including non-linear function optimization and neural network training, are proposed. The relationships between particle swarm optimization and both artificial life and genetic algorithms are described. A. Engelbrecht [3] provided a comprehensive introduction to the new computational paradigm of Swarm Intelligence (SI), a field that emerged from biological research and introduces the various mathematical models of social insects' collective behavior, and shows how they can be used in solving optimization problems.

III. PSO versus GA

3.1 Particle Swarm Optimization (PSO)

In this study, the basic PSO algorithm which is described in reference 7 is implemented. The basic algorithm is described first, followed by functional constraint handling, and finally, a discrete version of algorithm is presented.

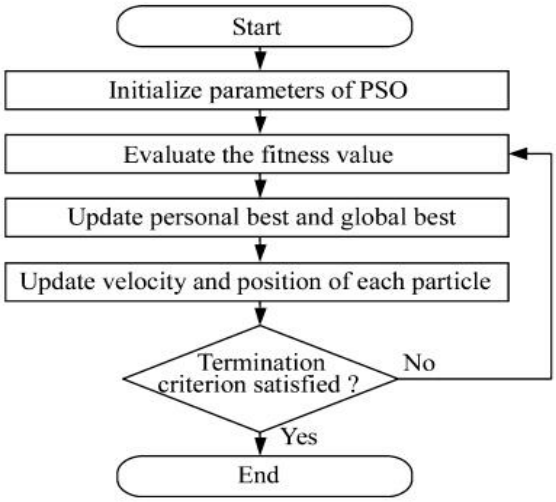

Figure 1: Flow chart of Particle Swarm Optimization

The GA encodes the design variables into bits of 0's and 1's, when it is inherently discrete. So, it is noted that it easily handles discrete design variables. Whereas, on the other hand, PSO is inherently continuous and should be modified to handle discrete design variables. The simple PSO algorithm consists of only three steps, namely, obtaining particle's positions and velocities, calculating velocity vectors and finally, position update.Consider a particle refers to a point in the design space which continuously changes its position for each iteration according to the velocity updates. Initially, the positions x_k^i and the velocities v_k^i of the initial swarm of particles are randomly generated in accordance with the upper and lower bounds on the design variable values, x_{min} and x_{max}as given in equations 1 and 2. The positions and velocities are expressed in vector format with the superscript and subscript denoting the i^{th} particle at time k. In the below equations 1 and 2, rn1 and rn2 are uniformly distributed random numbers between 0 and 1.

$x_0^i = x_{min} + rn1(x_{max} - x_{min})$	**(1)**
$v_0^i = \frac{x_{min} + rn2(x_{max} - x_{min})}{\Delta t}$	**(2)**

The next step is to calculate the velocities of all particles at time k+1 using the objective of the particles or the fitness values in the design space at time k. The fitness function value of .the particle states that which particle has the best global value in the current swarm, p_k^g and also determines best position of the particle, p^i.The velocity calculation formula uses this information along with the effect of current motion, v_k^i, to provide search direction and the next iteration will be v^i_{k+1}. The uniformly distributed variables rn1 and rn2 are used to prevent entrapment in local optima and to acquire a good coverage. Totally, there are three parameters to be used here. The first one is the inertia of the particle which is w and the other two parameters are c1 and c2. The c1 is the parameter which indicates that how much confidence it has in itself and c2 indicates that how much confidence it has in the swarm.

$v_{k+1}^i = wv_k^i + c1\ rn1\ \frac{(p^i - x_k^i)}{\Delta t} + c2\ rn2\ \frac{(p_k^g - x_k^i)}{\Delta t}$	**(3)**

In usual PSO algorithm, the values used for w, c1 and c2 are 1, 2 and 2 respectively. But, in this paper, it is found that by providing values such as 0.3, 1.3 and 1.2 to w, c1 and c2 respectively gives the best convergence rate for all kind of problems. Other combinations of values may produce slower convergence or no convergence at all.

Actual Pso Values		Modified Pso Values	
Weight Factors	Values	Weight Factors	Values
W	1	W	0.3
C1	2	C1	1.3
C2	2	C2	1.2

Table 1: Comparison of actual and modified PSO values

The last step is the position update. The position can be updated in each iteration in accordance with the velocity vector.

$$x_{k+1}^{i} = x_{k}^{i} + v_{k+1}^{i}\Delta t \qquad \textbf{(4)}$$

The velocity update, position update and fitness calculation are repeated until a desired convergence is obtained. In the PSO implemented in this study, the maximum changes in best fitness are the stopping criteria that should be smaller than the defined tolerance for a defined number of movesS, as in Equation 5. In this study, S is defined as ten moves and □is defined as for 10^{-5} all test problems.

$$f(p_{k}^{g}) - f(p_{k-q}^{g}) \leq \varepsilon \quad q = 1,2,.S \qquad \textbf{(5)}$$

In PSO, the design variable can take any value depending on their current position design space and the calculated velocity vector. When velocity vector increases rapidly the value of design variable can go outside their upper and lower limits,X_{min} or X_{max} .

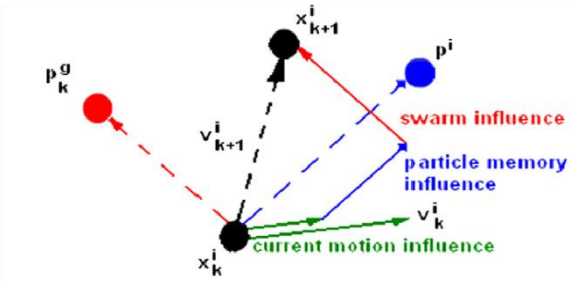

Figure 2: Illustration of velocity and position updates in Particle Swarm Optimization

This leads to a divergence. To avoid this problem in this paper, whenever the design variable violates their upper or lower design bounds, they are artificially brought back to their nearest side constraints. The functional constraints are handled in same way they are handled in GA. But in this comparison linear exterior penalty function is used to handle the function constraints as shown in Equation 6.

$$f(x) = \phi(x) + \sum_{i=1}^{N_{con}} r_i \max[0, g_{i(x)}] \qquad \textbf{(6)}$$

3.2 Genetic Algorithm (GA)

In this study, the benchmark problems can be solved by implementing basic binary encoded GA with tournament selection, uniform crossover and with lower probability mutation rate. The binary strings 0's and 1's referred to as chromosomes in GA is to represents the design variables of each individual design.GA works with coding for design parameters that permits both discrete and continuous parameters in one problem statement. To accomplish optimization-like process, the GA applies three operators to promulgate from one population to other population. The first operator is "Selection" operator that ridicules the principal of "Survival of the fittest". The second operator is "crossover"that ridicules mating in biological populations. The features good surviving design is promulgated from current population to the future population using crossover operator which will have better fitness value on average. The last operator is "mutation" that elevates diversity population characteristics.

The mutation operator permits a global search of design space and protects algorithm from getting trammeled in local minima.

3.2.1 Basic implementation of algorithm

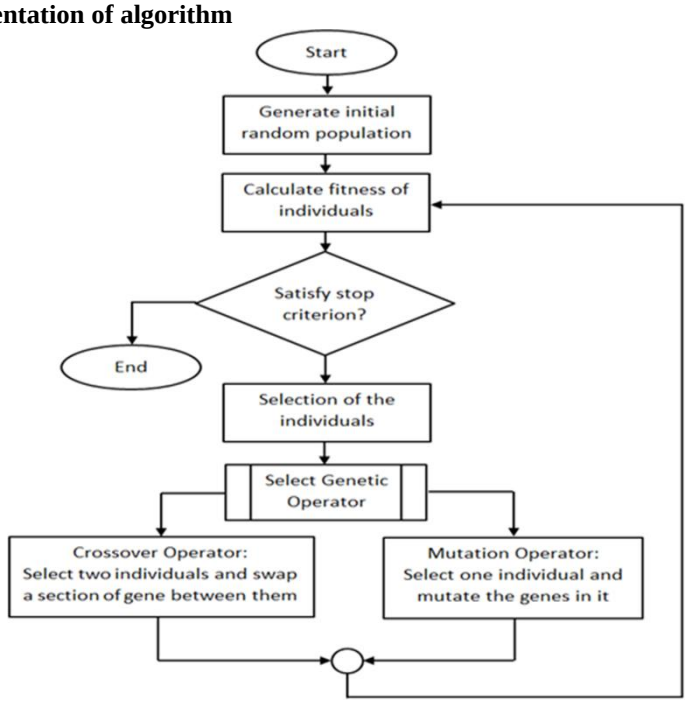

Figure 3: Flow chart of Genetic Algorithm

IV. Comparison Measure

The objective of this paper is to compare the performance of two empirical search methods GA and PSO using a set of test problems. The t-test (hypothesis test) is used to compare the effectiveness and efficiency of both the search algorithms.In this paper, two hypothesis are tested. The first test is the Effectiveness (Finding true global optimum) and the second test is the Efficiency (Computational cost) of the algorithms. The ability of the algorithm to find global solutions when the algorithm is started at the different random points in the design space. The solution's quality is measured by how familiar the solution is to the known global solution as shown in equation (7).

$$Q_{sol} \left| \frac{solution - known\ solution}{known\ solution} \right| \% \quad (7)$$

The solution quality measure described in equation (7) is used to composition a meaningful hypothesis to test the effectiveness test of search algorithms, PSO and GA.

4.1 Benchmark Test Problems: In this section a set of optimization benchmark testproblems are presented. These problems are solved by both GA and PSO. This set of problems include two, unconstrained and continuous functions like the banana(Rosenbrock) function and the Eggcrate function each has two design variables.

4.1.1 The Banana (Rosenbrock) Function: This function is known as the "Banana Function" because of its shape. The function is described mathematically in Equation (8). In this we have design variables with upper and lower limits of [-6 6] and it has a known global minimum of at [1 1] with optimal function value of zero.

$$Minimize\ f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \quad (8)$$

4.1.2 TheEggcrate Function: This function is described mathematically in equation(9). It has two design variables with upper and lower limits [-3π, 3π]. The eggcrate function has two known global minimum at[0 0] with an optimal function value of zero.

$$Minimize\ f(x) 25 + (sin^2 x_1 + sin^2 x_2) + x_1^2 + x_2^2 \quad (9)$$

V. Conclusion

Particle Swarm Optimization (PSO) is a relatively recent heuristic search method which is based on swarming and collaborative behavior in biological populations.The main objective of this paper is to test the hypothesis that states that although PSO and GA on average, yields the same effectiveness, but PSO is more computationally efficient than the GA. To examine this claim, two statistical tests were set to test the two elements of this claim, equal effectiveness but superior efficiency for PSO over the GA.The benchmark test problems included here are: the banana (Rosenbrock function) and the Eggcrate Function. For both PSO and GA, the effectiveness test can be done; it uses a quality of solution metric which measures the difference between the solutions obtained by the known solutions and the heuristic approaches of the test problems. The efficiency test uses the number of evaluation functions required by the search method to attain convergence. The same convergence criterion is enforced on both PSO and the GA. The results of the test that while both PSO and GA obtain high quality solutions, with quality indices of 99% for most test problems, to arrive at a high quality solutions, the computational effort required for the PSO is less than the effort required by the GA. So, the results again show the computational efficiency superiority of PSO over the GA is statistically proven.

References

[1]. J.H.Holland, -Genetic algorithms and the optimal allocation of trials, SIAM Journal of Computing, Vol.2, no.2, pp. 88-105, 1973.
[2]. Kennedy, J. and Eberhart, R., "Particle Swarm Optimization", Neural Networks, 1995.Proceedings of the IEEE International Conference on Neural Networks, Perth, WA, Australia , Vol. 4, pp. 1942-1948, 1995.
[3]. A.Engelbrecht, -Fundamentals of computational swarm intelligence, 2006, Hoboken: John Wiley & Sons, Ltd.
[4]. N.Nedjah, L. dos Santos Coelho, and L.de MacedoMourelle, - Multiobjective swarm intelligent systems: theory & experiences. Springer Science & Business Media, 2009, vol.261.
[5]. A.Kumar, A.Khosla, J.S.Saini and S.Singh,- Meta-heuristic range based node localization algorithm for wireless sensor networks, in localization and GNSS (ICLGNSS), 2012 International Conference on IEEE, 2012 , pp.1-7.
[6]. S.Singh, Shivangna and E. Mittal,-Range based wireless sensor nodes localization using PSO and BBO and its variants, in Communication Systems and Network Technologies (CSNT), 2013 International Conference on IEEE, 2013, pp. 309-315.
[7]. Venter, G. and Sobieski, J.,"Particle Swarm Optimization", AIAA 2002-1235,43rd IAAA /ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Denver, CO., April 2002.
[8]. Goldberg ,D., Genetic Algorithms in Search, Optimization and Machine Learning ,Addison –Wesley, Reading, MA,1989, pp. 1-25.
[9]. Williams, E.A., Crossley, W.A., "Emprically-derived population size and Mutation rate guidelines for a Genetic Algorithm with uniform crossover," Soft Computing in Engineering Design and Manfacturing, P.K. Chawdhry, R.Roy and R.K. Pant (Editors), Springer-Verlag, 1998,pp.163-172.
[10]. Harnett, D.,Statistical Methods, 3rd ed., Addison-Wesley, 1982,pp. 247-297.