International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

An Overview on Particle Swarm Optimization: Basic Concepts and Modified Variants

S. D. Chavan, Nisha P. Adgokar

[1]Professor, Department of Electronics & Telecommunication, Savitribai Phule, Pune University, D.Y.P.I.E.T., Pimpri, Pune, India

[2]Department of Electronics & Telecommunication, Savitribai Phule, Pune University, D.Y.P.I.E.T., Pimpri, Pune, India

Abstract: ***Particle Swarm Optimization (PSO) is stochastic optimization algorithm inspired by behavior of bird swarm searching for the food. PSO is a new, powerful intelligent swarm intelligence based algorithm used for finding optimum solution for complex problems. It can be modified to lots of other versions to increase speed of convergence and diversity. PSO variants are discovered to increase its performance and improve the ability to solve a wide range of optimization problems. In this paper, our focus is on classical PSO, its control parameters and modified versions.***

Keywords: Particle Swarm Optimization (PSO), Global Best PSO (GBPSO), Decreasing Weight PSO (DWPSO), Time-Varying Acceleration Coefficient PSO (TVACPSO), Guaranteed Convergence PSO (GCPSO), Adaptive PSO (APSO), Hybrid PSO (HPSO)

1. Introduction

The process used to find the best solution for a particular problem is called as optimization. Optimization techniques are of two types stochastic and deterministic. Solving optimization problems by using former methods leads to huge computational efforts and results in failure as the problem size increases. Reason behind popularity of bio-inspired stochastic algorithms is their computational efficiency as compared to the deterministic techniques. Swarm based algorithms, evolutionary algorithms are based on iterative improvement in solution and these techniques are called as meta-heuristic methods. Nature inspired algorithms has ability to solve complex problems from intrinsically very simple initial conditions and rules with some or no knowledge of the search space. Heuristic approach performs better to solve complex and difficult optimization problems specifically where traditional methods fail. Bio-inspired algorithm design formulation involves choosing proper problem representation, quality of solution evaluation by using fitness function and defining operators to produce new set of solutions. Bio-inspired optimization technique has two main categories i.e. Swarm based algorithms and evolutionary algorithms. Swarm based algorithms are inspired by collective behavior of animals and Evolutionary algorithms are designed on the natural evolution base [7].

In this paper we are considering Particle Swarm Optimization (PSO) technique which is based on Swarm Intelligence mechanism and inspired by behavior of swarms. It is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995 [6]. PSO is bio-inspired optimization technique which has many advantages like it is simple and easy to implement, computationally efficient and it has high convergence rate to get the best solution. It is an artificial intelligence (AI) technique that can be used to find approximate solutions to extremely difficult or impossible numeric maximization and minimization problems. PSO is based on the concept of multiple birds (particles) that search for the best food source (optimum) by using their inertia, their knowledge, and the knowledge of the swarm [13]. WSN has many issues such as node deployment, localization, energy-aware clustering and data-aggregation which are often formulated as optimization problems. For such problems, if we use traditional analytical optimization techniques then it will need vast computational efforts, and these efforts grow exponentially as the problem size increases. To avoid this, bio-inspired optimization methods are used as an alternative to analytical methods. Particle swarm optimization (PSO) is a popular multidimensional optimization technique which is considered as one of the competitive method to Genetic Algorithm. But PSO has many advantages over GA like PSO tends to converge to the best solution quickly i.e. PSO has high convergence rate as compared to GA. PSO is very simple and easy to implement. There are few parameters to adjust in PSO. It provides optimal solution to the applications where the computational resources such as memory and energy are extremely limited [1]. PSO is computationally efficient than GA. The number of particles (i.e., population size) is very small while for traditional evolutionary algorithms the population size is often required to be set as larger as possible to get an acceptable solution.

2. Classical Particle Swarm Optimization

2.1 Background

The term "Artificial Life" describes research into human-made systems that possess some of the essential properties of life. It includes:

1. Artificial life how computational techniques can help in the biological phenomena study.
2. Artificial life how biological techniques can contribute with computational problems.

Here, our focus is on this second topic in this report. Artificial neural network, ant colony optimization, genetic algorithm which is inspired by human evolution are some of the examples of this second aspect. In this paper we discuss specifically collective behavior of decentralized, self-organized systems interacting with their environment and each other which is known as swarm intelligence. Swarm intelligence is the discipline that deals with natural and artificial systems consist of many individuals that coordinate using decentralized control and self-organization. More specifically, the discipline focuses on the collective behaviors which result from the local interactions of the individuals with each other and with their environment. Examples of systems studied by swarm intelligence are colonies of ants, schools of fish, flocks of birds, herds of land animals. Some human artifacts are also included into this domain. Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) are the most popular examples of swarm intelligence. ACO is inspired by behavior of ants while PSO is based on behavior of bird swarm.

2.2 Introduction

The process of improving something than its current condition is called as optimization. It is the process of adjusting inputs, mathematical process or device characteristics to get the required output. This process is called as fitness function, objective function or cost function. Different methods are used for finding optimal solution while some of them are inspired by natural behaviour of some living elements like bees, birds, ants etc [5].

Particle Swarm Optimization is inspired by behaviour of bird flocking. This algorithm consists of swarm of particles i.e. group of random particles where each single solution is a bird (particle) in the search space. Optimized solution for every particle is determined by fitness function. Group of birds search for food by observing fitness function [10]. By following leader particle which is nearest to the food they can find the food. Leader particle is nothing but current optimal solution. So, every problem is initialized with random particles. PSO is based on birds swarm searching for optimal food sources in which direction of birds movement is influenced by its current movement, the best food source experienced by it ever and best food source any bird in the swarm ever experienced i.e. known as personal best and global best values and they get updated new best values after each iteration in PSO [5]. The personal best value is represented as u_p and global best value is represented as u_g. Particles movement is decided by following iteration in PSO:

$$u^{(i)}(n+1) = u^{(i)}(n) + v^{(i)}(n+1) \qquad (1.a)$$

$n = 0, 1, 2, \ldots, N-1,$

Where $u^{(i)}$ is the position of particle i, $v^{(i)}$ is the velocity of particle i, n is iteration, n= 0 indicates initialization and N is total number of iterations [2].

The velocity of the particle is given as

$$v^{(i)}(n+1)=v^{(i)}(n)+2r_1^{(i)}(n)[u_p^{(i)}(n)-u^{(i)}(n)] + 2r_2^{(i)}(n)[u_g(n) - u^{(i)}(n)], \qquad (1.b)$$

Where u_p is the personal best position, u_g is the personal best position. $u_p^{(i)}(n)-u^{(i)}(n)$ calculates vector in the direction of personal best position and $u_g(n) - u^{(i)}(n)$ gives vector directed towards global best position. $r_1^{(i)}$ and $r_2^{(i)}$ both represent random vectors which has values uniformly distributed between 0 and 1.

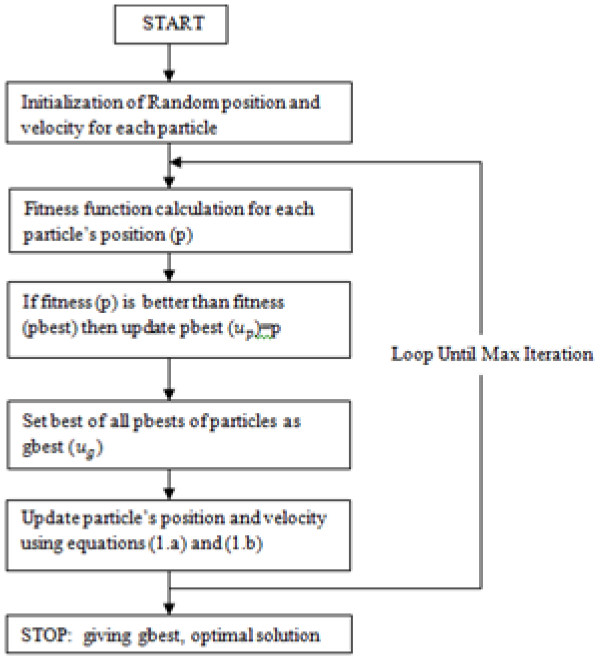

Figure 1: Flowchart of PSO Algorithm

As shown in flowchart in fig. (1), firstly initialization of swarm i.e. the position and velocity of particles are randomly initialized within the search space. After that, fitness function or objective values of particles are calculated. These first objective values and positions are automatically considered as personal best values and personal best positions. The best fitness value among all particles that particle's position and fitness value is considered as global best position and value in the entire swarm. By using equation (1.a) and (1.b) the position and velocity of particles are updated. In next iteration, again fitness value is calculated at updated position of each particle. Next step is to compare that personal best fitness with old personal best fitness value. If new fitness value is better than old one, we update personal best fitness value. Best of personal best value among all particles in whole swarm is set as global best fitness value. Then update particle's position and velocity according to above equations. This process continues with evaluating personal best positions and fitness values also global best position and value. This algorithm stops running when termination criterion meets i.e. either satisfactory optimal solution has been achieved or number of iterations is reached. Iteration stopping criteria can also be termination when no improvement is observed over a number of iteration.

2.3 The PSO Algorithm

As already discussed before, PSO is the simulation of the behaviors of bird flocking in which group of birds searches for food randomly in particular area. Initially they don't know the location of piece of food but after each iteration they will find out how far the food is available. They just follow the bird which is nearest to the food.

Each solution in PSO is "bird" in the search space which is known as particle. All these particles have fitness value which is evaluated from fitness function. Every particle has velocity which gives the flying direction of that particle in the search space. By following the current optimum particles, other particles fly through that problem space. Initialization is done with group of random particles (solution) firstly and then finding optimal solution at each iteration. The first solution is always considered as the best one. This best value is known as personal best u_p and the second best value which is known as global best value u_g is the best optimal solution obtained by any particle so far in the swarm. After that positions and velocities are updated from equations (1.a) and (1.b) respectively by using these two best values. This process will continue until last criteria of optimization or last iteration.

The pseudo code for PSO Algorithm is as follows:
I) For each particle:

Initialize particle
END

II) Do:
a) For each particle:
1) Determine fitness value.
2) If the fitness value is better than the personal best in the history.
3) Set current value as the new personal best value
END
b) For each particle:
1) Find in the particle neighborhood, the particle with the best fitness value among all particles in the whole swarm
2) Calculate particle velocity according to the velocity equation (1.b)
3) Apply the velocity constriction
4) Update particle position according to the position equation (1.a)
5) Apply the position constriction
END

While maximum iterations or minimum error criteria are not attained [8].

In the initialization step, the position in each dimension is initialized randomly. Velocities can be initialized randomly or set to 0. In the original algorithm, particles' velocities on each dimension are clamped to a maximum velocity Vmax. If the sum of accelerations would cause exceed in the velocity than Vmax , which is a parameter specified by the user, then the velocity on that dimension is limited to Vmax . This mechanism avoids the "swarm explosion" phenomenon. In practical applications, there is also a position constriction. The search space is bounded, so the particles' positions in each dimension have to be constrained inside those bounds.

2.4 Generic Algorithm (GA) Versus PSO

The procedure for most of the evolutionary algorithm is same as below:

1. Random generation of an initial population
2. Calculation of fitness value which will be directly proportional to the distance to the optimum.
3. Again reproduction of the population by using the fitness values.
4. If criteria of requirements achieved, then stop otherwise start from step 2 again.

So, above procedure is same for both Genetic Algorithm and PSO as both of them start with random initialization of population then fitness value calculation. Update the population based on fitness function and then last step. Genetic algorithm has genetic operators like crossover and mutation that PSO doesn't. Particles update themselves by using internal velocity values. The information sharing mechanism in PSO is different as compared to GA. In PSO, only global best gives out the information to others while in GA, chromosomes shares information with each other. It is one way information sharing mechanism in case of PSO.As compared to GA, all the particles tend to converge to best solution quickly in the local version of PSO.

3. Basic PSO Control Parameters

Particle Swarm Optimization technique design is based on two important factors: Exploration and exploitation. The ability of search algorithm to explore various search space to locate good optimum solution is known as exploration and the ability to concentrate the search around promising area to find out a solution is called as exploitation. There must be perfect balance between exploration and exploitation and in order to achieve this firstly quick identification of regions in search space instead of spending time in region which are already explored or which do not provide high quality solutions and secondly exploitation of collected search experience to locate optimal solution [12]. These two factors are very contradictory that's why trade-off between these two objectives must be achieved for optimal solution. Particle in swarm fly into the search space by their exploration and exploitation capabilities and use local best and global best positions to reach the best solution in PSO.

Variations in PSO are very necessary in order to improve speed of convergence and quality of solution. PSO has many control parameters like swarm size, inertia weight, neighborhood size, dimension of problem, number of iterations etc. These parameters are used to improve PSO performance [4].
Basic variants of PSO are as below:

3.1 Velocity Clamping

This phenomenon was introduced to avoid a phenomenon known as swarm explosion. With no restriction on the maximum velocity of the particles, a simple one-dimensional analysis of the swarm dynamic concludes that the particle velocity can grow unbounded while the particle oscillates around an optimum, increasing its distance to the optimum on each iteration. To control global exploration, velocity clamping is necessary. For that purpose, it's important to set some limit for the velocity of particle so that it will remain in the search area.

3.2 Inertia Weight

Inertia weight plays an important role in the process of providing balance between exploration and exploitation. Main purpose of inertia weight is to control the initial velocity. It determines contribution of previous velocity to the current step's velocity. Basic PSO doesn't have any inertia weight. The momentum of particle is controlled by it by weighting the contribution of previous velocity. So, after adding inertia weight to control velocity the equation (1.b) changes to

$$v^{(i)}(n+1)=wv^{(i)}(n)+c_1 r_1^{(i)}(n)[u_p^{(i)}(n)-u^{(i)}(n)]+ c_2 r_2^{(i)}(n)[u_g(n) - u^{(i)} (n)] \quad (2)$$

For n= 0,1, 2… N-1

Where w is inertia weight, c_1 and c_2 are the personal best and global best weights respectively [6]. Large value of inertia weight leads to global search and small value facilitates a local search.

If w is greater than one then the velocity will reduce with time and results in swarm divergence and in other case i.e. if w is less than one, the velocity of particle will reduce and at the end becomes zero. Exploration is supported by larger value of w and smaller value facilitates exploitation [11].

3.3 Constriction Coefficient

Velocity of the particle that using constriction coefficient is given as:

$$v^{(i)}(n+1)= x\{v^{(i)}(n)+ \emptyset_1[u_p^{(i)}(n)-u^{(i)}(n)] + \emptyset_2[u_g(n) - u^{(i)} (n)] \} \quad (3)$$

Where

$$x = \frac{2k}{\left|2-\emptyset-\sqrt{\emptyset(\emptyset-4)}\right|}$$

And $\emptyset = \emptyset_1 + \emptyset_2$

$\emptyset_1 = c_1 r_1$ and $\emptyset_2 = c_2 r_2$

Equation above is used under the constraints that $\emptyset \geq 4$ and $k \in [0,1]$. This is the dynamic technique for the convergence to the stable point convergence. After this, there is no need of velocity clamping. $\emptyset \geq 4$ and $k \in [0,1]$ these two conditions provide assurance to the convergence [4].

4. Modified Varients of PSO

The focus of Basic Particle Swarm Optimization can be either on convergence or diversity at any iteration. In diversity, particles are scattered, searching a large area and in case of convergence its meaning is that all particles are searching a small area, and all particles are close to each other. The best way is to focus on diversity in initial iterations and at later iterations concentration should be on convergence for best results.

4.1 Global Best Particle Swarm Optimization

Global best PSO (GBPSO) is one of the standard PSO variant. For finding velocity of the particle in this algorithm the equation is given as below:

$$v^{(i)}(n+1)=wv^{(i)}(n)+c_1 r_1^{(i)}(n)[u_p^{(i)}(n)-u^{(i)}(n)]+ c_2 r_2^{(i)}(n)[u_g(n) - u^{(i)} (n)], \quad (4)$$

Where n= 0, 1, 2….N-1,

w, c_1 and c_2 are inertia weight, personal best weight and global best weight respectively. These all are known as velocity weights. In classical PSO c_1 and c_2 are set to 2 and there is no inertial weight w as shown in equation (1.a) while in Global PSO c_1 and c_2 are not set to fixed value and inertial weight is used to control velocity parameter. The flow of this optimization technique is near about same as classical PSO as shown in fig. (1).

Step by step explanation of Global Best is as follows:

Step 1: Swarm initialization (Random initialization of positions and velocities in search space).

Step 2: Fitness value Calculation for each particle in the swarm.

Step 3: Compare the fitness value with personal best value of each particle, if current value is better, then update personal best value.

Step 4: Set best of all personal bests of all particles as a global best values.

Step 5: Update velocities and positions of all particles.

Step 6: Check for the required criterion met or not. If yes then terminate whole algorithm otherwise again start with step 2.

4.2 Decreasing Weight Particle Swarm Optimization

Decreasing weight PSO (DWPSO) is same as the Global best PSO in all manners except that inertial weight is decreased continuously with time. DWPSO concentrates on diversity in initial iterations and on convergence in later iterations. This is the best strategy for getting better results. So, here the velocity equation from (4) changes to

$$v^{(i)}(n+1)= w(n)v^{(i)}(n)+c_1 r_1^{(i)}(n)[u_p^{(i)}(n)-u^{(i)}(n)]+ c_2 r_2^{(i)}(n)[u_g(n) - u^{(i)} (n)] \quad (5)$$

n= 0, 1, 2, ….N-1

Where inertia weight w at every iteration n is determined by using following equation:

$$w(n) = w_f - (w_f - w_l)\frac{n}{N} \quad (6)$$

Where $w(n)$ is the inertia weight at iteration n,

w_f stands for inertia weight for the first iteration and

w_l represents inertia weight for the last iteration N.

The inertia weight w depends upon the value of N i.e. total number of iterations in this algorithm. Behavior of this algorithm at every iteration can be changed by changing the total number of iterations N. To restart an algorithm from certain point is not possible, if N is changed [5].

4.3 Time-Varying Acceleration Coefficient PSO

In this variant of PSO, all velocity weights which are inertia weight w, c_1 and c_2 which are personal best and global best weights vary over time. In this TVACPSO algorithm also the

main aim is to achieve a high diversity at starting iterations and a high convergence for ending iterations. The inertia weight changes as in DWPSO by equation (6) [5].
Calculation of velocity is done by using following equation:

$$v^{(i)}(\mathrm{n+1}) = w(n)v^{(i)}(\mathrm{n}) + c_1(\mathrm{n})r_1^{(i)}[u_p^{(i)}(n) - u^{(i)}(\mathrm{n})] + c_2(\mathrm{n})r_2^{(i)}[u_g(\mathrm{n}) - u^{(i)}(\mathrm{n})], \quad (7)$$

N= 0, 1, 2….N-1,
Personal best and global best weights are given as follows:

$$c_1(\mathrm{n}) = c_{1f} - (c_{1f} - c_{1l})\frac{n}{N} \quad (8.a)$$

$$c_2(\mathrm{n}) = c_{2f} - (c_{2f} - c_{2l})\frac{n}{N} \quad (8.b)$$

c_1(n) and c_2(n) are the personal best weight and global best weight at iteration n respectively. c_{1f} and c_{2f} are personal best and global best weights designed for first iteration. c_{1l} and c_{2l} are the personal best weight and global best weight designed for last iteration N respectively.

4.4 Guaranteed Convergence PSO

Guaranteed Convergence PSO (GCPSO) technique focuses on search within dynamically adapted radius by global best particles for all time. The main drawback is the immobilism or stagnation but it increases local convergence by using the global best particle for random search in a radius which adaptively changes at all times. In this technique, velocities and inertia weights are determined by using equation (5) and (6) respectively for each particle.c_1 and c_2 are kept constant in this technique. To update global best particle's position following iteration is used.

$$u^{(i_g)}(\mathrm{n+1}) = -u^{(i_g)}(\mathrm{n}) + u_g(\mathrm{n}) + w(\mathrm{n})v^{(i_g)}(\mathrm{n}) + p(n)(1-2r_3(\mathrm{n})) \quad (9)$$

N= 0,1,2…..N-1
Where i_g represents index of the latest global best particle, $-u^{(i_g)}$(n) + u_g(n) is the expression used to reset position of the particle i_g to the global best position. r_3(n) represents random number uniformly distributed between 0 and 1. p represents search radius parameter used to control search radius. It is calculated by using:

$$p(\mathrm{n+1}) = \begin{cases} 2p(\mathrm{n}) & \text{if } \check{s}(\mathrm{n+1}) > s_c, \\ \frac{1}{2}p(\mathrm{n}) & \text{if } \check{f}(\mathrm{n+1}) > f_c, \\ p(\mathrm{n}) & \text{otherwise,} \end{cases} \quad (10)$$

Where s_c is the success threshold and f_c is the failure threshold. Success means when equation (9) results in global best position and value and if not achieved then it results in failure. To calculate consecutive success and failure value following equations can be used:

$$\check{s}(\mathrm{n+1}) = \begin{cases} 0 & \text{if } \check{f}(\mathrm{n+1}) > \check{f}(\mathrm{n}), \\ \check{s}(\mathrm{n}) + 1 & \text{otherwise,} \end{cases} \quad (11.a)$$

$$\check{f}(\mathrm{n+1}) = \begin{cases} 0 & \text{if } \check{s}(\mathrm{n+1}) > \check{s}(\mathrm{n}), \\ \check{f}(\mathrm{n}) + 1 & \text{otherwise,} \end{cases} \quad (11.b)$$

4.5 Adaptive Particle Swarm Optimization

Adaptive PSO (APSO) has many variants available based on variations in some or all w, c_1 and c_2 velocity weights. For dynamically adaptive PSO, there are two factors required. First one is evolutionary speed factor which is used to changes in personal best value and second one is an aggregation degree that computes the relative position of particles in the objective space to determine value of inertia weight w. Some APSO adapts inertia weight according to its fitness value, global best and global worst value. Self- tuning APSO algorithm is all about self-adaptation of personal best weights, global best weights and inertia weights. When positions of most of the particles in whole search space stops changing for large number of successive iterations though the global optimum remains undiscovered, this situation is known as premature convergence. This happens because of the small inertia weight.

To reduce premature convergence another version of APSO is introduced in which inertial weight changes based on swarm diversity. The swarm diversity is determined as a function of positions. By using this technique overall convergence will get increased [11].

Four states APSO is another type of APSO which includes four states, exploration, exploitation, convergence and jumping-out based on the velocity weights c_1 and c_2. When the current value of personal best weight c_1 increases and global best weight c_2 decreases then first state of exploration occurs. Exploitation state occurs when there is slight increase in c_1 and slight decrease in c_2 . It enters in convergence state if both weights c_1 and c_2 increases slightly and jumping out state is a result of decrease in c_1 and increase in c_2. The state selection in every iteration is done by using evolutionary factor which is computed by using mean of the separation of particles in the swarm. Dynamic optimization problems can be solved by using Detection and Response APSO by monitoring global best and second best positions. In this technique, change in fitness function occurs because of the change in their fitness value in any two iteration which is a result of position change [5].

In this way there are many types of APSO which provides better search efficiency than classical PSO. The main advantage is that its convergence speed of global search over whole search space. The performance of PSO can be increased by many factors like global optimality, solution accuracy and algorithm reliability by using APSO techniques.

4.6 Hybrid Particle Swarm Optimization

Hybrid optimization is the technique in which two or more optimization techniques are combined. When PSO is combined with one or more optimization techniques then it is called as hybrid PSO. In the initial stages of a global search the convergence speed is more in PSO but search process will slow down around global optimum. So, combination of PSO and Back-Propagation (BP) algorithm can be used as a hybrid version in order to get high convergence speed as well as convergence accuracy. In another hybrid PSO the advantages of Nelder-Mead (NM) Simplex method and PSO are combined [5]. This algorithm provides balance of PSO's global searching and SM's accurate local search. This approach has robust, high convergence rate, precision and it can give satisfactory solutions of nonlinear equations.

Similarly combination of different bioinspired optimization techniques can be used to provide better results. For global optimization of multimodal functions hybrid version of Genetic Algorithm (GA) and Particle Swarm Optimization which is known as GA-PSO algorithm. Though PSO has many advantages over GA, main problem of PSO is premature convergence [3]. Hybrid version of PSO and GA is used to overcome limitations of PSO. This hybrid algorithm has three approaches. In first one approach global best particle position does not change its position for some steps. The crossover operation is performed on global best particle with chromosomes of GA. PSO and GA both run in parallel in this type. In second type mutation operator of GA is used to change positions of the stagnated personal best particles. In the third model of this hybrid version total number of iterations are divided equally among GA and PSO. First half of total iterations is used by GA and solution of GA is assigned as initial population to PSO algorithm. Similarly, Ant Colony Optimization and PSO combination is used as another hybrid PSO known as ACO/PSO technique used to optimize multicast tree. In this algorithm large number of mobile agents generated initially, their movement is guided by pheromones as in ACO and global maximum of attribute values are obtained through the random interaction between agents using PSO algorithm [9].

5. Conclusion

In the past several years, Particle Swarm Optimization (PSO) has been successfully applied in many research and application areas such as fuzzy system control, function optimization, artificial neural network training, wireless sensor network etc. It is proved that PSO gets better results in faster, cheaper way as compared to other optimization method. PSO is very popular optimization technique because it has very few parameters to adjust. For wide variety of application, classical PSO can be modified to another version with slight variation in parameter.

References

[1] Erfu Yang, Ahmet T. Erdogan, Tughrul Arslan, Nick Borton, “An Improved Particle Swarm Optimization Algorithm For Power Efficient Wireless Sensor Networks,” ECSIS Symposium on Bio-inspired, Learning, and Intelligent Systems for Security, 2007.

[2] Raghvendra. Kulkarni, Ganesh Kumar Venayagamoorthy,“Particle Swarm Optimization in Wireless Sensor Networks: A Brief Survey,” IEEE, 2010.

[3] Seyedali Mirjalili, Siti Zaiton Mohd Hashim, “A New Hybrid PSOGSA Algorithm for Function Optimization,” International Conference on Computer and Information Application (ICCIA 2010). 2010.

[4] Siti Mariyam Shamsuddin, Siti Sophiyati Yuhani, Dian Palupi Rini, “Particle Swarm Optimization: Technique, system and challenges,” International Journal of Computer Applications (0975 – 8887) Volume 14– No.1, January 2011.

[5] Thomas Schoene, “Step-Optimized Particle Swarm Optimization, “A Thesis Submitted to the College of Graduate Studies and Research in the Department of Computer Science University of Saskatchewan,Saskatoon,” August 2011.

[6] J. C. Bansal, P. K. Singh, Mukesh Saraswat, Abhishek Verma, Shimpi Singh Jadon, Ajith Abraham, “Inertia Weight Strategies in Particle Swarm Optimization,” IEEE, 2011.

[7] Binitha S, S Siva Sathya, “A Survey of Bio inspired Optimization Algorithms,” International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-2, May 2012.

[8] Reza Toushmalani, “Gravity inversion of a fault by Particle swarm optimization (PSO),” SpringerPlus 2013, 2013.

[9] Manoj Kumar Patel, Manas Ranjan Kabat, Chita Ranjan Tripathy, “A hybrid ACO/PSO based algorithm for QoS multicast routing problem,” Ain Shams Engineering Journal, Elsevier, 2013.

[10] Shahram Jamali, Leila Rezaei, Sajjad Jahanbakhsh Gudakahriz, “An Energy-efficient Routing Protocol for MANETs: a Particle Swarm Optimization Approach,” Journal of Applied Research and Technology Vol. 11, December 2013.

[11] Kaushik Suresh, Sayan Ghosh, Debarati Kundu, Abhirup Sen, Swagatam Das and Ajith Abraham, “Inertia-Adaptive Particle Swarm Optimizer for Improved Global Search,” Center of Excellence for Quantifiable Quality of Service, Norwegian University of Science and Technology, Trondheim, Norway.

[12] Yenny Noa Vargas, Stephen Chen, “Particle Swarm Optimization with Resets – Improving the Balance between Exploration and Exploitation,” University of Havana, Havana, Cuba, School of Information Technology, York University, Toronto, ON M3J 1P3 Canada.

[13] Devi Manickavelu, Rhymend Uthariaraj Vaidyanathan, “Particle swarm optimization (PSO)-based node and link lifetime prediction algorithm for route recovery in MANET,” Eurasip journal of wireless communications and networking, 2014.

Author Profile

Nisha P. Adgokar received the B.E degree in Electronics and Telecommunications Engineering from Saint Gadgebaba University, Amravati, Maharashtra, India in 2008. She is now pursuing Master degree in communication engineering from D.Y.P.I.E.T. Pimpri, Pune, India along with working as visiting lecturer in D.Y. Patil University, Pune.

Mr. S. D. Chavan, working as Associate Professor M. E. (Electronics), PhD (Appeared) Electronics. Experience consists of 13 years of Academics and 5 years of Industrial experience. Professional Body membership include LMISTE, MIETE