Review of Particle Swarm Optimization Techniques

Anuradha L. Borkar
Electronics and Telecommunication Dept
M S S S' College of Engineering and Technology
Jalna, Maharashtra, India
anu1may@rediffmail.com

***Abstract*— Particle Swarm Optimization (PSO) is a population-based, self-adaptive search optimization technique. PSO attracts researchers around the world due to its powerful searching ability and simplicity. PSO is computationally faster than other optimization technique and requires very less memory for implementation. Considerable researches have been carried out to improve the performance of PSO. This paper reviews the development of the particle swarm optimization method in recent years. Modifications to adapt to different and complex environments are reviewed.**

Keywords: **Particle Swarm Optimization (PSO), Linearly Decreasing Weight PSO (LDW-PSO), Supervisor Student Model in PSO (SSM-PSO), PSO with Time Varying Acceleration Coefficients (PSOTVAC) Global Local Best PSO (GLBest PSO) Parameter Free PSO algorithm (pf-PSO) PSO with extrapolation technique (e-PSO)**

I. INTRODUCTION

Optimization problems are frequently encountered in many engineering, economic or scientific fields that engineers or researchers are seeking to minimize cost or time, or to maximize profit, quality or efficiency, of a specific problem. In addition, evolutionary Optimization techniques are extensively used in engineering to optimize multiobjective, multimodal, multidimensional, nonlinear function and real word problems. The Genetic algorithm (GA) ruled several years for solving complex real world problems [1-2]. Apart from the modification of GA, new paradigm have been developed i.e. Particle Swarm Optimization (PSO). In the past decade, PSO algorithm attracts researchers around the world due to its powerful searching ability and simplicity. PSO simulates the swarm behavior of birds flocking and fish schooling that swarms work in a collaborative manner to search for foods as efficient and quick as possible[1-2]. Particle swarm optimization is a population-based, self-adaptive search optimization technique. The PSO is a global optimization technique and it is has been successfully applied to image processing, pattern recognition, video compression, antenna design etc. Particle swarm optimization is very simple concept and can be implemented in a few lines of computer code.

This paper reviews the development of the particle swarm optimization method in recent years. This paper also aims to gain an insight into some of the design issues of particle swarm optimization through reviewing the different mechanisms of utilizing local search information and the various techniques to achieve a balance between exploration and exploitation. Section 2 depicts the term related to PSO. The section 3 depicts the original Particle Swarm Optimization (PSO). The section 4 depicts the new variants of PSO. Section 5 comprises of conclusion.

II. TERMS REALTED TO PSO

Swarm: Population or particles or group of particles.
Particle: Member (Individual) of the swarm. Each particle represents a potential solution to the problem being solved. The position of the particle is determined by the solution it currently represents.
p_{best} *(p_i):* Personal or local best position of a given particle, so far.
g_{best} *(p_g):* Global best position of the best particle member of the neighborhood of a given particle .
Velocity (V): This vector determines the direction in which a particle needs to move (fly) in order to improve current position.
Inertia Weight (w): It controls current velocity of particle by impact of previous history.
Learning Factor or Acceleration Constant (C_1 and C_2): c_1 is the cognitive learning factor and represents the attraction that a particle has toward its own success c_2 is the social learning factor and represents the attraction that a particle

.

III. STANDARD PARTICLE SWARM OPTIMIZATION

The original framework of PSO is designed by Kennedy and Eberhart in 1995[3]. There fore it is known as standard PSO. It follows the optimization process by means of personal or *local best* (p_i), *global best* (p_g), particle position or displacement (X) and particle velocity (V). For each particle, at the current time step, a record is kept for the position, velocity, and the best position is found in the search space. Each particle memorizes its previous velocity and the previous best position and uses them in its movements. Each particle has position and velocity. Velocity is the rate of change of position.
The position of *i*-th particles are represented by Eq. (1).

$$X= x_1, x_2, x_3 \ldots xi. \tag{1}$$

and velocity is given by Eq. (2)

$$V= v_1, v_2, v_3 \ldots v_i \tag{2}$$

Particles change their positions by flying around in a multidimensional search space. For each particle, at the current time step or generations (t), a record is kept for the position, velocity, *local best* and *global best* positions. The best previous positions (p_{best} or *local best*) of any particle is recorded are represented by Eq. (3)

$$Pi= p_1, p_2, p_3 \ldots pi \tag{3}$$

The best particle positions among all the particle in population is known as *global best* and represented by $p_{g.}$ The fitness or objective function is used to observe the performance of each particle. Each particle in the swarm is represented by following characteristics.
X_i = *Current position of particle*
V_i= *Current velocity of particle*
P_i= *Personal or local best position of the particle or local best*
P_g = *Global best position among all particles*
The two updating fundamental equations in a PSO are velocity and position equations, which are expressed as Eq (4) and (5). The velocities of the particles are limited in [*Vmin Vmax*]^D. If smaller than *Vmin*, an element of the velocity is set equal to *Vmin* or *Xmin*. If greater than *Vmax*, and then equal to *Vmax* or *Xmax*. Since the original version of PSO lacks velocity control mechanism, it has a poor ability to search at a fine grain

$$V_{id}(t+1) = V_{id}(t) + c_1 * r_{1d}(t) * (p_{id}(t) - X_{id}(t)) + c_2 * r_{2d}(t) * (p_{gd}(t) - X_{id}(t)) \tag{4}$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \tag{5}$$

Where, t= Current iteration or generation.
i = Particle Number
d= Dimensions
$V_{id}(t)$ = Velocity of i-th particle for d-dimension at iteration number t .
$X(t)$ = Position of i-th particle for d-dimension at iteration number t.
C_1 and C_2= Acceleration constants
$r_{1d}(t)$ and $r_{2d}(t)$ = Random values [0 1] for d- dimension at iteration number t
$p_{id}(t)$= Personal or local best i-th particle for d-dimension at iteration number t
$p_{gd}(t)$ = Global best for d-dimension at iteration number t

Refer the Eq.(4), the right side of which consists of three parts. The first part of equation is the previous velocity of the particle. The second part is the cognition (self-knowledge) or memory, which represents the particle, is attracted by its own previous best position and moving toward to it. The third part is the social (social knowledge) or cooperation, which represents the particle, is attracted by the best position so far in population and moving toward to it. There are restrictions among these three parts and can be used to determine the major performance of the algorithm. The details of variables used in PSO and their importance in improving the performance of PSO are given below.

IV. NEW VARIANTS OF PSO

Computational time is bottleneck of many optimization techniques. Another bottleneck to optimization algorithm is the problem of creating infeasible solution when computational time required is more. The high computational cost and accuracy of global solution are forced the researchers to develop efficient optimization techniques in terms of new, modified or hybrid soft computing techniques. PSO has been paid more and more attention by researchers.

Most of the works focus on two aspects. The first is the performance improvement of PSO by modifying parameters, increasing population diversity and hybrid with other optimizing approaches. The second is the applications of PSO in different areas, such as multi-objective optimization, electronics, training neural network, security and mute selection of network, medicine and emergent system identification, etc. The performance of the PSO can be improved by using self adaptive parameters. In this case, the PSO parameters are adapted based on the feedback from previous search. The another approach for improvement of PSO performance is the hybridization of PSO with other optimization technique.

. These modifications in PSO will not guarantee the best feasible solution. Still on date, researchers are trying to improve PSO performance in terms of convergence and accuracy. Many researchers have shown that the algorithm can be improved by adding features to it or making it more complicated. The new or improved algorithms of PSO are proposed by many researchers by making modifications in standard PSO which are given below.

A. Modified Particle Swarm Optimization

The Yuhui Shi and Russell proposed a modified particle swarm optimization in 1998[4]. The inertia weight (*w*) brought into equation of velocity for balancing global and local search. The position equation remains same. Without *w* in velocity equation is a known as simple PSO and with *w* it is known as modified PSO. The two updating equations in a modified PSO are velocity and position equations, which are expressed as Eq (6) and (7).

$$V_{id}(t+1) = w * V_{id}(t) + c_1 * r_{1d}(t) * (p_{id}(t) - X_{id}(t)) + c_2 * r_{2d}(t) * (p_{gd}(t) - X_{id}(t)) \tag{6}$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \tag{7}$$

The range of inertia weight *w* decides convergence to global best solution. A large inertia weight favors exploration (global search) while small inertia weight favors exploitation (local search).

B. Linearly Decreasing Weight PSO (LDW-PSO)

Eberhart and Shi introduced a time decreasing inertia weight over the generations to overcome the disadvantage of modified PSO [5]. By linearly decreasing the inertia weight from a large value to small value, the PSO tends to have more global search ability at the beginning of the run while having more local search ability near the end of the run. The time varying inertia weight (TVIW) is given by Eq. (8).

$$w = (w_1 - w_2)\left(\frac{\max iter - iter}{\max iter}\right) + w_2 \tag{8}$$

Where w_1 and w_2 are the higher and lower inertia weight. Current generation is denoted by "*iter*" and "*maxiter*" is the total number of generations. The suggested value $w_1 = 0.9$ and $w_2 = 0.4$ will improve the performance of PSO.

C. Supervisor Student Model in Particle Swarm Optimization (SSM-PSO)

In traditional PSO algorithms, particles have a chance to fly out of the search space $[Xmin, Xmax]^D$ resulting in invalid solutions .To bring the particles with a defined search space, the velocity and positions are particles are limited in $[Vmin, Vmax]^D$ and $[Xmin, Xmax]^{D.}$ *Vmin* is usually equal to *Xmin***,** and *Vmax* equal to *Xmax*
Yu Liu and Zheng Qin introduced Supervisor Student Model in Particle Swarm Optimization (SSMPSO) [5] in 2004. SSM-PSO can prevent particles from flying out of defined region without checking the validity of positions at every iteration. It reduces computational cost in two aspects.

1. It introduces a new parameter, called momentum factor, into the position update equation, which can restrict the particles inside the defined search space without checking the boundary at every iteration.
2. On the other hand, *Relaxation- Velocity-Update strategy* that is to update the velocities of the particles as few times as possible during the run.

In SSM-PSO, particles are manipulated according to the following Eq (9) and (10).

$$V_{id}(t+1) = V_{id}(t) + c_1 * r_{1d}(t) * (p_{id}(t) - X_{id}(t)) + c_2 * r_{2d}(t) * (p_{gd}(t) - X_{id}(t)) \tag{9}$$

$$X_{id}(t+1) = (1 - mc) * X_{id}(t) + mc * V_{id}(t+1) \tag{10}$$

Where *mc* is momentum factor ($0 < m < 1$), and *Vmin=Xmin*: *Vmax=Xmax*. The suggested value of *mc* is 0.3 for better performance of PSO**.** In traditional PSO, velocity of particles is updated at every iteration but in SSMPSO, velocity of each particle will updated if fitness of particles at current generation is not better than the fitness of particles at previous generation otherwise velocity will not be changed. The SSPPSO reduces computational cost by avoiding frequently updating the velocity of particles**.**

D. Particle Swarm Optimization with Time Varying Acceleration Coefficients (PSOTVAC)

Asanga Ratnaweera and Saman K. Halgamuge introduced Particle Swarm Optimization with Time Varying Acceleration Coefficients (PSOTVAC)[6] in 2004 to control the local search and convergence to the global optimum solution. It is used to avoid premature convergence in the early stages of the search and to enhance convergence to the global optimum solution during the latter stages of the search. Kennedy and Eberhart described that a relatively high value of the cognitive component, compared with the social component, will result in excessive wandering of individuals through the search space. In addition, high value of the social component may lead particles to rush prematurely toward a local optimum. The suggested values of the acceleration coefficients at 2, in order to make the mean of acceleration coefficients are unity. Due to this particles would over fly only half the time of search. The fixed value of acceleration coefficients at 2 generate better solutions instead of linearly decreasing both acceleration coefficients with time.
The new strategy of time varying acceleration coefficients (TVAC) is used to

1. To encourage the individuals to wander through the entire search space, without clustering around local optima, during the early stages of the optimization.
2. During the latter stages, it is very important to enhance convergence toward the global optima, to find the optimum solution efficiently.
3. To enhance the global search in the early part of the optimization and to encourage the particles to converge toward the global optima at the end of the search.

The time varying acceleration coefficients c_1 and c_2 reduce the cognitive component and increase social component. The large cognitive component and small social component result in to move the particle around the search space. A small cognitive component and large social component result in the global optimum at the end. With a large cognitive component and small social component at the beginning, particles are allowed to move around the search space, instead of moving toward the population best. On the other hand, a small cognitive component and a large social component allow the particles to converge to the global optima in the latter part of the optimization. The time varying acceleration coefficients (TVAC) is given by Eq. (11) and (12).

$$c_1 = (c_{1i} - c_{1f})\left(\frac{\max iter - iter}{\max iter}\right) - c_{1f} \tag{11}$$

$$c_2 = (c_{2i} - c_{2f})\left(\frac{\max iter - iter}{\max iter}\right) - c_{2f} \tag{12}$$

Where c_{1i} and c_{2i} are the initial values of the acceleration coefficients c_1 and c_2. c_{1f} and c_{2f} are the final values of the acceleration coefficients c_1 and c_2. The suggested value of c_1 from 2.5 to 0.5 and c_2 from 0.5 to 2.5 results in good performance of PSO.

E. Global Local Best Particle Swarm Optimization (GLBest PSO)

The GLBestPSO [7] is introduced by Senthil Arumugam et. al. in 2005. In GLBestPSO, the inertia weight (*w*) and acceleration coefficient (c_1 and c_2) are neither set to a constant value nor set as a linearly decreasing time varying function. Instead, they proposed in terms of global best (*gbest*) and local best (*pbest)* position of the particles. The GLBest inertia weight (GLbest TW) and GLBest acceleration co-efficient (GLbest AC) are given in Eq. (13) and (14).

$$GLBestIW = w_i = \left(1.1 - \frac{gbest_i}{(pbest_i)_{average}}\right) \tag{13}$$

$$GLBestAC = c = \left(1 + \frac{gbest_i}{(pbest_i)}\right) \tag{14}$$

The modified velocity equation for the GLBest PSO is given in Eq. (15). The position equation of GLBest PSO is same as other PSO methods.

$$V_i(t) = w * V_i(t-1) + c_i * r(t) * (pbest_i + gbest_i - 2X_i(t)) \tag{15}$$

F. Parameter Free PSO algorithm (pf-PSO)

Senthil Arumugam et. al introduced new PSO algorithm known as Parameter free PSO in 2007[8]. In Pf-PSO, local best (*pbest*) and global best (*gbest*) positions are used to update the positions of particles. The main advantage of Pf-PSO algorithm is that it does not need a velocity equation. In addition, it dose not require any parameters like inertia weight and acceleration coefficient for tuning to reach global best position. The positions of particles are updated by following Eq.(16)

$$X_i(t) = [1 - {gbest}/{X_i(t-1)}] * Rnd_1 * gest + [\frac{gbest}{X_i(t-1)}] * Rnd_2 * pbest_i \tag{16}$$

Where,
gbest= Global best
pbest= Local best
Rnd_1 and Rnd_2 = Random functions between [0 1]

G. Particle swarm optimization with extrapolation technique (e-PSO)

M. Senthil Arumugam M.V.C. Rao, Alan W.C. Tan introduced epos [8] (PSO with extrapolation technique) in 2009 for solving optimization problems with low computational cost. This method uses extrapolation technique to perform the optimization; hence it is named as extrapolated PSO or shortly ePSO algorithm. The current particle position and the global best particle position are involved in the extrapolation operation. In ePSO, the current particle position is updated by extrapolating the global best particle position and the current particle positions in the search space in order to refine the search towards the global optimum value. The two extrapolation co-efficient (e_1 and e_2) which are much similar to the time-varying inertia weight (TVIW). The values of extrapolation co-efficient are linearly decreasing with the generations from 0.999 to 0.36. The updated position equation for each particle is given in Eq. (17). The position equation is formulated with the global best (gbest) position, local best position (pbest) and the current position of the particle.

$$X_i(t) = [gbest] + [e_1 * Rnd * gbest] + [e_1 * (gbest - X_i(t-1)) * \exp(e_2 * ((f(gbest) - f(X_i(t-1))) / f(gbest)))] \tag{17}$$

Where e_1 and e_2 are extrapolation co-efficient, $e_1 = e_2 = exp$ *(-current generation/max. no. of generation)*. *Rnd* is the random function which varies between (0, 1). A Random function is also included to incorporate the stochastic behavior into the algorithm's search to find the optimal solution. *gbest* is the particle position where the best fitness solution is found, $X_i(t)$ is the current particle position in *t*-th generations. *f (gbest)* is the fitness value at *gbest* position *f (Xi(t))* is the current particle's fitness value.

The global best position of the particle is the main operator for extrapolation. The second term ensures the movement of the current particle towards the gbest position and the third term gives the step size for the movement for the updated position. During the initial stage of the simulation, the difference between the fitness value at gbest position *f(gbest)* and the current particle's fitness value *f(Xi(t))* is high, so the particle will move more distance towards the gbest position and the difference will be reduced in later stages since the current particle will be close to the gbest position. When the difference between *gbest* and *X* or *Xi (t)* is more then the distance between *gbest* position and extrapolated positions is also relatively more. During the early stages of the simulation, the step size will be large and thus the position of *X* is lying away from *gbest* position. During the final stage of the simulation, the current particle's fitness will become closer to the *gbest* particle's fitness value and hence the step size is reduced to smaller value. So the current particle's position will come closer to the *gbest* position. The main advantage of ePSO algorithm is that it does not need a velocity equation. The position of the particle is updated directly by extrapolating the current particle position with the global best particle position obtained so far.

V. CONCLUSION

In his paper the author reviews the development of the particle swarm optimization method in recent years. The most of researchers tried to improve the performance to PSO by changing the inertia weight and acceleration coefficients which balances between local and global search. The inertia weight (*w*) is used for balancing global and local search. Due to linearly decreasing inertia weight from a large value to small value, the PSO tends to have more global search ability at the beginning of the run while having more local search ability near the end of the run. Momentum factor restrict the particles inside the defined search space without checking the boundary at every iteration. The time varying acceleration coefficients reduce the cognitive component and increase social component. The inertia weight and acceleration coefficient proposed in terms of global best (*gbest*) and local best (*pbest)* position of the particles to avoid tuning of parameters. The main advantage of Pf-PSO algorithm is that it does not need a velocity equation and it dose not require any parameters like inertia weight and acceleration coefficient for tuning to reach global best position. In ePSO, the current particle position is updated by extrapolating the global best particle position and the current particle positions in the search space in order to refine the search towards the global optimum value. Therefore above review of PSO helps to researchers new variants of PSO.

REFERENCES

[1] R. C. Eberhart and Y. Shi, "Comparison between Genetic Algorithm and Particle Swarm Optimization," *IEEE International Conference* on *Computational Intelligence,* Anchorage, AK, May 1998, pp. 611-616.

[2] J. Kennedy, "The particle swarm: Social adaptation of knowledge," in Proc. IEEE Int. Conf. Evolutionary Computation, 1997, pp. 303–308.

[3] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *IEEE International Conference on Neural Networks*, Perth, Australia. vol. 4, Dec. 1995, pp. 1942-1948

[4] Y. Shi, R.C. Ebcrhart, "A modified particle swarm optimizer," Proceedings of the IEEE Congress on Evolutionary Computation (CEC 1998), Piscataway, NI. pp. 69-73, 1998.

[5] Yu Liu, Zheng Qin, Xingshi He, "Supervisor-Student Model in Particle Swarm Optimization", IEEE Congress on Evolutionary Computation 2004 (CEC 2004), vol.1, pp. 542-547, June 2004.

[6] Ratnaweera, S. Halgamuge, and H. Watson, "Self Organizing Hierarchical Particle Swarm Optimization with time varying acceleration coefficients", IEEE transactions on Evolutionary Transactions, vol. 8, pp. 240-255, 2004.

[7] M.Senthil Arumugam , M.V.C.Rao, Aarthi Chandramohan, " A new and improved version of particle swarm optimization algorithm with global-local best parameters", Journal of Knowledge and Information System (KAIS), Springer, vol.16, no.3, pp.324-350, 2008

[8] M.Senthil Arumugam , M.V.C.Rao, Alan W.C. Tan, " A new novel and effective particle swarm optimization like algorithm with extrapolation technique", International Journal of Applied Soft Computing, Elsevier, vol.9, pp. 308-320, 2009