International Journal of Allied Practice, Research and Review
Website: www.ijaprr.com (ISSN 2350-1294)

Study on Particle Swarm Optimization

Om Prakash[1] and Sajal Kumar Das[2]

Abstract - Particle swarm optimization is a stochastic, population-based computer problem-solving algorithm; it is a kind of swarm intelligence that is based on social- principles and provides insights into social behavior, as well as contributing to social-psychological engineering applications. The aim of this paper is to give fundamental insight into the particle swarm optimization algorithm

Keywords: ***Particle swarm optimization, Evolutionary computing, inertia weight***

I. Introduction

The last three decades have witnessed the development in efficient and effective stochastic optimizations. In contrast to the traditional adaptive stochastic search algorithms, evolutionary computation (EC) techniques exploit a set of potential solutions, namely a population, and detect the optimal solution through cooperation and competition among the individuals of the population. These techniques often detect optima in difficult optimization problems faster than traditional methods [3]. One of the most powerful swarm intelligence-based optimization techniques, named PSO, was introduced by Kennedy and Eberhart [1, 2]. PSO is inspired by the swarming behavior of animals, and human social behavior. During the last decade many studies focused on this method and almost all of them, strongly confirmed the abilities of this newly proposed optimization technique [1, 3, 4, 7], e.g. fast convergence, finding global optimum in presence of several local optima, simple programming and adaptability with constrained problems. Some author attempted to enhance the algorithm by developing new variations such as variable inertia coefficient, constriction factor [4], maximum velocity limit, parallel optimization, deflection, repulsion, stretching [2], mutation [7,8] etc. Particle swarm optimization was invented by Russ Eberhart and James Kennedy in 1995 through simplifying a social simulation model which was originally developed to simulate the process of birds seeking food. The PSO algorithm is a population-based evolutionary algorithm. Like other evolutionary algorithms, each individual (called particle in PSO) in the population represents a candidate solution to the problem to be solved. Unlike other evolutionary

* Asstt. Prof., ECE Department, JJT University, Jhunjhunu (Raj.)
** Modam R&D, Ericsson (I) PVT Ltd., Outer Ring Road,Bangalore

algorithms each individual/particle has a velocity parameter associated with it in addition to its position parameter in the solution space, which is the only parameter that an individual in other evolutionary algorithms has. Each particle "flies" through the solution space with a velocity which is dynamically changed according to its own flying experience and its companion's flying experience. It is this velocity changing rule through which all the particles communicate and share information among themselves. Furthermore, it is this sharing and communicating mechanism that enables particles to fly towards better and better search areas while at the same time to risk to be stuck into local minima.

The search process or flying trajectories of particles are complicated and nonlinear. To search for good enough solutions, especially for the multi-modal optimization problems, the search process needs to have the ability to converge at some time while diverging at other times in order to have the ability to find good enough solutions and to be able to avoid to be stuck in un-wanted local minima. Therefore, it is critical to have a capability to monitor the search process of PSO in order to first understand the PSO search process and then design a better algorithm or even have possibilities to control the search process later.

A straightforward approach to measure the diversity of PSO is to use the standard deviation of the fitness values of all the population particles. Population fitness values are attributes of the PSO behaviors and not the PSO particles themselves directly. Therefore, this kind of diversity measurement is simple but it is an indirect measurement of the population diversity. The diversity of PSO has been looked at from different perspectives. Each particle in a PSO has an *n*-dimensional velocity associated with it in addition to its position as in other evolutionary algorithms. Therefore, diversities depend on particles' positions and velocities instead of only the position diversity as in other evolutionary algorithms. Velocity diversity has velocity speed diversity and velocity directional diversity. The velocity speed tells how fast a particle is flying and the velocity direction tells where a particle is flying towards [6].

II. Particle swarm optimization algorithms

The Particle swarm optimization algorithm is an optimization and search technique based on the principles of social behavior of animals. The method was developed in 1995 by James Kennedy and Russell Eberhart. PSO mimics the collective intelligent behavior of "unintelligent" creatures. PSO is very good at finding good enough solutions for a large range of problems, such as constrained optimization problems, multi-objective optimization problems, etc. The original PSO algorithm is very simple in concept and easy in implementation.

Initialization

The initial swarm is generally created with all particles randomly distributed throughout the design space, each with a random initial velocity vector. Eq. (1) is used for obtaining the random initial position and Eq. (2) for velocity vector, it can be formulated as:

$$x_{id}^{(0)} = x_i^{min} + r_1\left(x_i^{max} - x_i^{min}\right);\ i = 1,2,\dots,n;\quad d = 1,2,\dots,m \tag{1}$$

$$v_{id}^{(0)} = \frac{x_i^{min} + r_2\left(x_i^{max} - x_i^{min}\right)}{\Delta t};\quad i = 1,2,\dots,n;\quad d = 1,2,\dots,m. \tag{2}$$

where,

- $x_{id}^{(0)}$ represents the *d*th position value of the *i*th particle at time step is zero.

- $v_{id}^{(0)}$represents the rate of the *d*th position value change (velocity) for particle *i*at time step is zero.
- $r_1$and r_2 are random number within the range of [0,1].
- x_i^{min} is lower bound of the position.
- x_i^{max} is upper bound of the position.
- Δt is step size.
- *n*is the number of variables.
- *m* is the size of the swarm (number of particle in swarm).

Parameters of particle swarm optimization are also initializing.c_1 and c_2 are positive constants, known as thrust parameter. Generalized value of c_1 and c_2 is 2 and $c_1 + c_2 \leq 4$. $w$is the inertia of the particles. Upper and lower bounds are usually specified on v_i to avoid too rapid movement of particles in the search space; that is, the various range of the *dth* velocity is [-Vmax, Vmax]. In this thesis upper and lower bonds are formulated as:

$$v_i^{min} = -\alpha\, x_i^{min};\ \ i = 1,2,\dots,n. \qquad (3a)$$
$$v_i^{max} = \alpha\, x_i^{max};\ i = 1,2,\dots,n. \qquad (3b)$$

where,
- v_i^{min} lower bond of velocity.
- v_i^{min} upper bond of velocity.
- $\alpha$is the arbitrary constant with in the range [0,1], generally taken as 0.5 in this thesis. For x_i^{min}, if x_i^{min} is positive then α is positive and vice versa. For x_i^{max} α is always positive.
- x_i^{min} is lower bound of the position.
- x_i^{max} is upper bound of the position.

Updating position and velocity

New velocity and Position can be updated using Eqn. (4) and Eqn. (5) respectively. These equations are formulated as [5]:

$$v_{id}^{(t+1)} = wv_{id}^{t} + c_1 r_3\left(x_i^{lb} - x_{id}^{t}\right) + c_2 r_4(x_i^{gb} - x_{id}^{t} ;i = ,2, \dots n\, ; d = 1,2, \dots m.$$
(4)

where
- c_1 and c_2 are positive constants,
- $r_3$and r_4 are random number within the range of [0,1].
- $w$is the inertia of the particles.
- $x_{id}^{(t)}$represents the *d*th position value of the *i*th particle at time step *t.*
- x_i^{lb}represents the *d*th position value of the best previous position (the position giving the best fitness value) of the *i*th particle at the time step *t.*
- x_i^{gb} represents the index of the best particle among all the particles.
- $v_{id}^{(t)}$represents the rate of the *d*th position value change of the *i*th particle at time step *t*.
- $v_{id}^{(t+1)}$ represents the rate of the *d*th position value change (velocity) for particle *i*at time step *t+1*

$$x_{id}^{(t+1)} = x_{id}^{t} + v_{id}^{(t+1)}, i = 1,2, \dots n; d = 1,2, \dots m \qquad (5)$$

where,

- $x_{id}^{(t)}$represents the *d*th position value of the *i*th particle at time step *t*.
- $x_{id}^{(t+1)}$represents the *d*th position value of the *i*th particle at time step *t+1*.
- $v_{id}^{(t+1)}$represents the rate of the *d*th position value change (velocity) for particle *i*at time step *t+1*.
- *n*is the number of variables.
- *m* is the size of the swarm (number of particle in swarm).

Usually, all the w will have the same value for simplicity but the inertia weight can be dynamically adjusted according to the current and historical performance of the particles, which will improve the PSO's performance since the search process of a PSO algorithm is nonlinear and complicated. A simple and straightforward approach is to linearly decrease inertia weigh over the course of PSO. Other PSO parameters can be fixed and/or even can be dynamically changed to affect the search process in the hope of having a more diverse or better performed PSO particles.

Equation (4) and Eqn. (5) are the equations governing the flying trajectory of particles and tells change of the velocity. In order not to violate the physical law, the velocity cannot be changed abruptly and shall be changed from the current velocity, which is reflected by the first part of the Eq. (4) as a "flying" particle's momentum. The other two parts of the Eq. (4) reflect the learning and collaboration capability of a particle. The second part reflects a particle's self-learning capability or self-cognition, that is, a particle learns from its own flying experience. The third part reflects particle's collaboration capability, that is, a particle learns from "flying" experience of its neighboring particles. The position of a "flying" particle is adjusted according to the Eq. (5).

There are two most commonly used versions of PSOs, global version and local version. In a global version PSO, a single and unique global best is shared by all particles in the whole population. In a local version PSO, each particle in the population may have different global best which is the best performed particle within the particle's own neighborhood. In both global and local version PSO, particles fly through the search space with dynamically changed velocities according to the Eq. (4). The neighborhood of each particle is generally defined as its topologically nearest particles at each side instead of Euclidean neighborhood. The global version PSO can be considered as a special case of a local version PSO if the whole population is considered as each particle's neighborhood. It has been claimed that the global version PSO converges fast, but with potential to converge to the local minimum, while the local version PSO might have more chances to find better solutions slowly.

Inertia weight

The inertia weight, w, controls the momentum of the particle by weighing the contribution of the previous velocity–basically controlling how much memory of the previous flight direction will influence the new velocity. The equation of inertia weight is formulated as

$$W = \frac{(w^{max} - w^{min}) * IT}{IT^{max}} \qquad (6)$$

where

- wis the inertia of the particle.
- w^{max} represents upper limit of inertia weight.
- w^{min} represents lower limit of inertia weight.
- ITrepresents current number of iteration.
- IT^{max} represents maximum number of iteration.

The inertia weight w is in the range [0.4, 0.9] and declines linearly in iteration as described in equation (6). For w > 1, velocities increase over time causing divergent behavior. Particles fail to change direction in order to move back towards promising areas. For w < 0, particles decelerate until their velocities reach zero.

Convergence criterion

Changes in the objective function are monitored for a specified number of consecutive design iteration. If the maximum change in the objective function is less than a predefined allowable change, convergence is assumed.Evolution flowchart is shown below in Fig. 1.

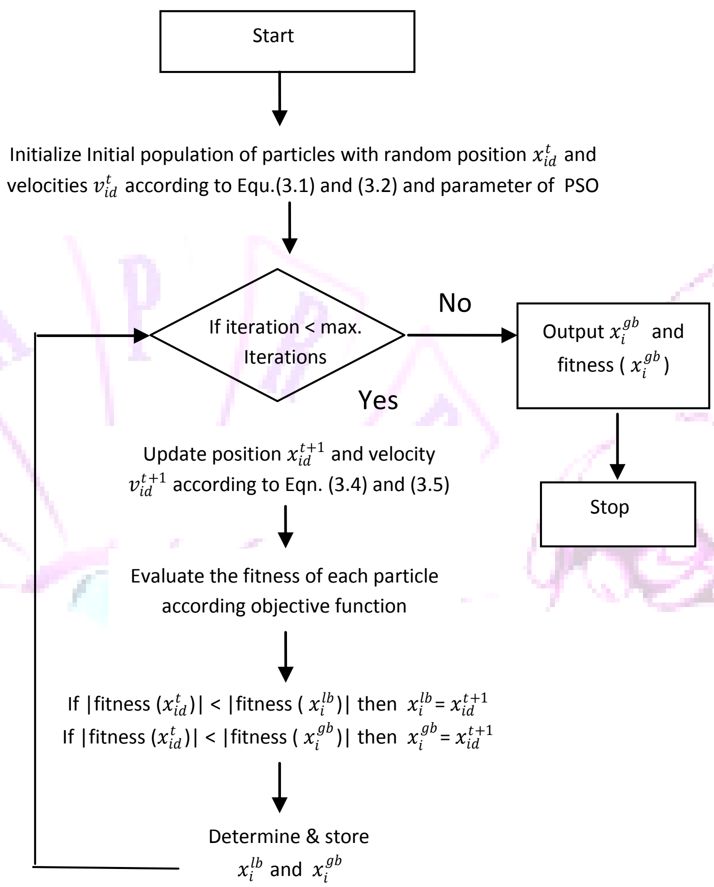

Fig. 1 Evaluation flowchart of PSO

III. REFERENCES

[1] Eberhart R and Kennedy J. [1995], A new optimizer using particle swarm theory, Proceedings of 6th Symposium on MicroMachine and Human Science, Nagoya, Japan, pp. 39–43.

[2] Kennedy, J., Eberhart, R. C. and Shi, Y. [2001], Swarm intelligence. San Francisco: Morgan Kaufmann Publishers.

[3] Parasopoulos, K.E., Vrahatis, M.N. [2004]. On the computation of all global minimizers through particle swarm optimization. IEEE Transactions on Evolutionary Computation vol. 8, no. 3, pp.211-224.

[4] Parsopoulos, K. E., and Vrahatis, M. N. [2002], Particle swarm optimization method for constrained optimization problems. In P. Sincak, J. Vascak, V. Kvasnicka & J. and Pospichal (Eds.), intelligent technologies–theory and application (New trends in intelligent technologies). Frontiers in artificial intelligence and applications, vol. 76, pp. 214–220. Amsterdam: IOS Press, ISBN:1-58603-256-9.

[5] Shi, Y., and Eberhart, R.C. [1998], Parameter selection in particle swarm optimization. In: Porto, V.W., Saravanan,N.,Waagen, D., Eiben, A.E. (eds.) Evolutionary Programming, vol. 7, pp. 591–600. Springer-Verlag,Berlin.

[6] Shi Y. and Eberhart R. [2009], Monitoring of particle swarm optimization, Front. Computer science, China, vol. 3, no. 1, pp. 31-37.

[7] Tan, R.R. [2007], Hybrid evolutionary computation for the development of pollution prevention and control strategies. Journal of Cleaner Production vol. 15, no. 10, pp. 902–906.

[8] Zhang .J, Wang .J,and Yue .C; [2012]Small Population Based Particle Swarm Optimization for short term Hydrothermal Scheduling. IEEE transaction on Power System, vol.27, no.1, pp.142-152.