

A general framework of particle swarm optimization

Loc Nguyen

Independent scholar, Vietnam

Email: ng_phloc@yahoo.com

Homepage: www.locnguyen.net

Abstract

Particle swarm optimization (PSO) is an effective algorithm to solve the optimization problem in case that derivative of target function is inexistent or difficult to be determined. Because PSO has many parameters and variants, I propose a general framework of PSO called GPSO which aggregates important parameters and generalizes important variants so that researchers can customize PSO easily. Moreover, two main properties of PSO are exploration and exploitation. The exploration property aims to avoid premature converging so as to reach global optimal solution whereas the exploitation property aims to motivate PSO to converge as fast as possible. These two aspects are equally important. Therefore, GPSO also aims to balance the exploration and the exploitation. It is expected that GPSO supports users to tune parameters for not only solving premature problem but also fast convergence.

Keywords: global optimization, particle swarm optimization (PSO), exploration, exploitation.

1. Introduction to particle swarm optimization (PSO)

Particle swarm optimization (PSO) algorithm was developed by James Kennedy (a social psychologist) and Russell C. Eberhart (an electrical engineer). This section is navigated by the article “Particle swarm optimization: An overview” of Riccardo Poli, James Kennedy, and Tim Blackwell. The main idea of PSO is based on social intelligence when it simulates how a flock of birds search for food. Given a target function known as *cost function* $f(\mathbf{x})$, the optimization problem is to find out the minimum point \mathbf{x}^* known as minimizer or optimizer so that $f(\mathbf{x}^*)$ is minimal. In PSO theory, $f(\mathbf{x})$ is also called *fitness function* and thus, when $f(\mathbf{x})$ is evaluated at $f(\mathbf{x}_0)$ then, $f(\mathbf{x}_0)$ is called fitness value which represents the best food source for which a flock of birds search.. If \mathbf{x}^* is an optimizer, $f(\mathbf{x}^*)$ is called optimal value, best value, or best fitness value. As a convention, the optimization problem is global minimization problem when \mathbf{x}^* is searched over entire domain of $f(\mathbf{x})$. For global maximization, it is simple to change a little bit our viewpoint.

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} f(\mathbf{x})$$

Traditional local optimization methods such as Newton-Raphson and gradient descent along with global optimization methods require that $f(\mathbf{x})$ is differentiable. Alternately, PSO does not require existence of differential. PSO scatters a population of candidate solutions (candidate optimizers) for \mathbf{x}^* and such population is called swarm whereas each candidate optimizer is called particle in the swarm. PSO is an iterative algorithm running over many iterations in which every particle is moved at each iteration so that it approaches the global optimizer \mathbf{x}^* . Movement of all particles is attracted by \mathbf{x}^* . In other words, such movement is attracted by minimizing $f(\mathbf{x})$ so that $f(\mathbf{x})$ is small enough. In PSO, \mathbf{x} is considered as position of particle. The movement of each particle is affected by its best position and the best position of the swarm. Note, the closer to \mathbf{x}^* , the better the position is.

As a formal definition, let \mathcal{S} be the swam of particles and let \mathbf{x}_i and \mathbf{p}_i be current position and best position of particle i . Note, \mathbf{p}_i is called *local best position*. Moreover, the movement speed of particle i is specified by its velocity \mathbf{v}_i . Let \mathbf{p}_g be the *global best position* of entire swarm. The closer to \mathbf{x}^* , the better the positions \mathbf{p}_i and \mathbf{p}_g are. It is expected that \mathbf{p}_g is equals to \mathbf{x}^* or is approximated to \mathbf{x}^* . The ultimate purpose of PSO is to determine \mathbf{p}_g .

$$\mathbf{p}_g \cong \underset{\mathbf{x}}{\operatorname{argmin}} f(\mathbf{x})$$

Of course, \mathbf{x}_i , \mathbf{p}_i , and \mathbf{p}_g are n -dimensional points and \mathbf{v}_i is n -dimensional vector because $f(\mathbf{x})$ is from real n -dimensional space \mathbf{R}^n to real space \mathbf{R} . Following is pseudo-code of PSO (Wikipedia, 2017).

<p>Input: the swarm \mathcal{S} of particles along with their initialized positions and velocities. Output: the global best position \mathbf{p}_g of entire swarm with expectation that \mathbf{p}_g is equal or approximated to the global minimizer \mathbf{x}^*.</p> <p>Let \mathbf{lb} and \mathbf{ub} be lower bound and upper bound of particles in their search space. They are vectors. All current positions \mathbf{x}_i of all particles are initialized randomly. Moreover, their best positions are set to be their current positions such that $\mathbf{p}_i = \mathbf{x}_i$. Note, all particles are randomized in the range $[\mathbf{lb}, \mathbf{ub}]$ as closed sphere.</p> $\mathbf{p}_i = \mathbf{x}_i \in [\mathbf{lb}, \mathbf{ub}]$ <p>All current velocities \mathbf{v}_i of all particles are initialized randomly. Because each \mathbf{v}_i is vector, its elements are randomized in the range $[- \mathbf{ub} - \mathbf{lb} , \mathbf{ub} - \mathbf{lb}]$ from $- \mathbf{ub} - \mathbf{lb}$ to $\mathbf{ub} - \mathbf{lb}$ where the notation \cdot denotes distance between two vectors or two points.</p> $\mathbf{v}_i = \begin{pmatrix} v_{i1} \\ v_{i2} \\ \vdots \\ v_{in} \end{pmatrix} \text{ where } v_{ij} \in [- \mathbf{ub} - \mathbf{lb} , \mathbf{ub} - \mathbf{lb}]$ <p>The global best position \mathbf{p}_g is assigned by the local best position \mathbf{p}_i such that $f(\mathbf{p}_i)$ is smallest among particles.</p> $\mathbf{p}_g = \underset{\mathbf{p}_i}{\operatorname{argmin}} f(\mathbf{p}_i)$ <p>While terminated condition is not met do For each particle i in swarm \mathcal{S} Velocity of particle i is updated as follows: $\mathbf{v}_i = \mathbf{v}_i + U(0, \phi_1) \otimes (\mathbf{p}_i - \mathbf{x}_i) + U(0, \phi_2) \otimes (\mathbf{p}_g - \mathbf{x}_i) \quad (1.1)$ Position of particle i is updated as follows: $\mathbf{x}_i = \mathbf{x}_i + \mathbf{v}_i \quad (1.2)$ If $f(\mathbf{x}_i) < f(\mathbf{p}_i)$ then The best position of particle i is updated: $\mathbf{p}_i = \mathbf{x}_i$ If $f(\mathbf{p}_i) < f(\mathbf{p}_g)$ then The best position of swarm is updated: $\mathbf{p}_g = \mathbf{p}_i$ End if End if End for End while</p>
--

Table 1.1. Basic particle swarm optimization (PSO) algorithm

Equation 1.1 is the heart of PSO, which is called *velocity update rule*. Equation 1.2 is called *position update rule*. There are two most popular terminated conditions:

1. The cost function at \mathbf{p}_g which is evaluated as $f(\mathbf{p}_g)$ is small enough. For example, $f(\mathbf{p}_g)$ is smaller than a small threshold.
2. Or PSO ran over a large enough number of iterations.

Function $U(0, \phi_1)$ generates a random vector whose elements are random numbers in the range $[0, \phi_1]$. Similarly, function $U(0, \phi_2)$ generates a random vector whose elements are random numbers in the range $[0, \phi_2]$. For example,

$$U(0, \phi_1) = (r_{11}, r_{12}, \dots, r_{1n})^T \text{ where } 0 \leq r_{1j} \leq \phi_1$$

$$U(0, \phi_2) = (r_{21}, r_{22}, \dots, r_{2n})^T \text{ where } 0 \leq r_{2j} \leq \phi_2$$

Note, the super script “ T ” indicates transposition operator of vector and matrix. The operator \otimes denotes component-wise multiplication of two points (Poli, Kennedy, & Blackwell, 2007, p. 3). For example, given random vector $U(0, \phi_1) = (r_{11}, r_{12}, \dots, r_{1n})^T$ and position $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{in})^T$, their component-wise multiplication is:

$$U(0, \phi_1) \otimes \mathbf{x}_i = \begin{pmatrix} r_{11}x_{i1} \\ r_{12}x_{i2} \\ \vdots \\ r_{1n}x_{in} \end{pmatrix}$$

Two components $U(0, \phi_1) \otimes (\mathbf{p}_i - \mathbf{x}_i)$ and $U(0, \phi_2) \otimes (\mathbf{p}_g - \mathbf{x}_i)$ are considered as attraction forces that push every particle to move. Sources of force $U(0, \phi_1) \otimes (\mathbf{p}_i - \mathbf{x}_i)$ and force $U(0, \phi_2) \otimes (\mathbf{p}_g - \mathbf{x}_i)$ are the particle i itself and its neighbors. Thus, two most important parameters of PSO are ϕ_1 and ϕ_2 which represent the two attraction forces. The popular values of them are $\phi_1 = \phi_2 = 1.4962$. Parameter ϕ_1 along with the force $U(0, \phi_1) \otimes (\mathbf{p}_i - \mathbf{x}_i)$ express the *exploitation* of PSO whereas parameter ϕ_2 along with the force $U(0, \phi_2) \otimes (\mathbf{p}_g - \mathbf{x}_i)$ express the *exploration* of PSO (Poli, Kennedy, & Blackwell, 2007, p. 4). The larger parameter ϕ_1 is, the faster PSO converges but it trends to converge at local minimizer. In opposite, if parameter ϕ_2 is large, convergence to local minimizer will be avoided in order to achieve better global optimizer but convergence speed is decreased. Parameters ϕ_1 and ϕ_2 are also called acceleration coefficients or *attraction coefficients*. Especially, ϕ_1 is called *cognitive weight* and ϕ_2 is called *social weight* because ϕ_1 reflects thinking of particle itself in moving and ϕ_2 reflects influence of entire swarm on every particle in moving. In practical, velocity \mathbf{v}_i can be bounded in the range $[-\mathbf{v}_{max}, +\mathbf{v}_{max}]$ in order to avoid out of convergence trajectories but the parameter \mathbf{v}_{max} is not popular because there are some other parameters such as inertial weight and constriction coefficient (mentioned later) which are used to damp the dynamics of particles. Favorite values for the size of swarm (the number of particles) are ranged from 20 to 50.

Because any movement has inertia, inertial force is added to the two attraction forces. Hence, the inertial force is represented by a so-called *inertial weight* ω where $0 < \omega \leq 1$. Equation 1.1 becomes (Poli, Kennedy, & Blackwell, 2007, p. 4):

$$\mathbf{v}_i = \omega \mathbf{v}_i + U(0, \phi_1) \otimes (\mathbf{p}_i - \mathbf{x}_i) + U(0, \phi_2) \otimes (\mathbf{p}_g - \mathbf{x}_i) \quad (1.3)$$

The larger inertial weight ω is, the faster particles move because its inertial is high, which leads PSO to explore global optimizer. Note that moving fast does not imply fast convergence. In opposite, the smaller ω leads PSO to exploit local optimizer. In general, large ω expresses exploration and small ω expresses exploitation. The inverse $1-\omega$ is known as friction coefficient. The popular value of ω is 0.7298 given $\phi_1 = \phi_2 = 1.4962$.

Pioneers in PSO (Poli, Kennedy, & Blackwell, 2007, p. 5) recognized that if velocities \mathbf{v}_i of particles are not restricted, their movements can be out of convergence trajectories at unacceptable levels. Therefore, they proposed a so-called *constriction coefficient* χ to damp dynamics of particles. Note, χ is also called *constriction weight* or *damping weight* where $0 < \chi \leq 1$. With support of constriction coefficient, equation 1.1 becomes (Poli, Kennedy, & Blackwell, 2007, p. 5):

$$\mathbf{v}_i = \chi \left(\mathbf{v}_i + U(0, \phi_1) \otimes (\mathbf{p}_i - \mathbf{x}_i) + U(0, \phi_2) \otimes (\mathbf{p}_g - \mathbf{x}_i) \right) \quad (1.4)$$

It is easy to recognize that equation 1.3 is special case of equation 1.4 when the expression $\chi \mathbf{v}_i$ is equivalent to the expression $\omega \mathbf{v}_i$. The popular value of constriction coefficient is $\chi = 0.7298$ given $\phi_1 = \phi_2 = 2.05$ and $\omega = 1$. Note, inertial weight ω is also the parameter that damps dynamics of particles. This is the reason that $\omega = 1$ when $\chi \neq 1$ but constriction of χ is stronger than ω because χ affects previous velocity and two attraction forces whereas ω affects only previous velocity.

Structure of swarm which is determined by defining neighbors and neighborhood of every particle is called *swarm topology* or *population topology*. Because \mathbf{p}_g is the best position of entire swarm, the attraction force $U(0, \phi_2) \otimes (\mathbf{p}_g - \mathbf{x}_i)$ indicates the movement of each particle is affected by all other particles, which means that every particle connects to all remaining particles. In other words, neighbors of a particle are all other particles, which is known as fully connected swarm topology. For easily understandable explanation, suppose particles are vertices of a graph, fully connected swarm topology implies that such graph is fully connected, in which all vertices are connected together. Alternately, swarm topology can be defined in different way so that each particle i only connects with a limit number K_i of other particles. In other words, each particle has only K_i neighbors. With custom-defined swarm topology, equation 1.4 is written as follows (Poli, Kennedy, & Blackwell, 2007, p. 6):

$$\mathbf{v}_i = \chi \left(\mathbf{v}_i + \frac{1}{K_i} \sum_{k=1}^{K_i} U(0, \phi) \otimes (\mathbf{q}_k - \mathbf{x}_i) \right) \quad (1.5)$$

Where \mathbf{q}_k is the best position of the k^{th} neighbor of particle i . Of course, \mathbf{q}_k is \mathbf{p}_j of some particle j .

$\mathbf{q}_k = \mathbf{p}_j$ such that particle j is the k^{th} neighbor of particle i .

Please pay attention that, in equation 1.5, particle i is also its neighbor. In other words, in equation 1.5, the set of K_i neighbors includes particle i . The two parameters ϕ_1 and ϕ_2 are reduced into only one parameter $\phi > 0$, which implies the strengths of all attraction forces from all neighbors on particle i are equal. The popular value of ϕ is $\phi = 2.05$ given $\chi = 0.7298$. Equation 1.5 is known as Mendes' fully informed particle swarm (FIPS) method. The topology in the basic PSO specified by equation 1.1, equation 1.3, and equation 1.4 is known *global best topology* because only one best position \mathbf{p}_g of entire swarm is kept track. However, equation 1.5 indicates that many best positions from groups implied by neighbors are kept track. Hence, FIPS specifies a so-called *local best topology*, which converges slowly but avoids converging at local optimizer. In other words, local best topology aims to exploration rather than exploitation. However, at a compromise, FIPS makes convergence speed of PSO slow because the exploitation is scarified for the exploration. Therefore, I propose a general framework of PSO in next section which aims to balance the exploration and the exploitation.

2. General PSO with probabilistic constriction coefficient

Recall that the two main aspects of PSO are exploration and exploitation. The exploration aspect aims to avoid premature converging so as to reach global optimizer whereas the exploitation aspect aims to motivate PSO to converge as fast as possible. Besides exploitation property can help PSO to converge more accurately regardless of local optimizer or global optimizer. These two aspects are equally important. Consequently, two problems corresponding to the exploration and exploitation are *premature problem* and *dynamic problem*. Solutions of the premature problem are to improve the exploration and solutions of the dynamic problem are to improve the exploitation. Inertial weight and constriction coefficient are common solutions for dynamic problem. Currently, solutions of dynamic problem often relate to tuning coefficients which are PSO parameters. Solutions of premature problem relates to increase dynamic ability of particles such as:

- Dynamic topology.
- Change of fitness function.
- Adaptation includes tuning coefficients, adding particles, removing particles, and changing particle properties.
- Diversity control.

The proposed general framework of PSO called *GPSO* aims to balance the exploration and the exploitation, which solves both premature problem and dynamic problem. If we focus on the fact that the attraction force issued by the particle i itself is equivalent to the attraction force from the global best position \mathbf{p}_g and the other attraction forces from its neighbors \mathbf{q}_k , equation 1.5 is modified as follows:

$$\mathbf{v}_i = \chi \left(\omega \mathbf{v}_i + U(0, \phi_1) \otimes (\mathbf{p}_i - \mathbf{x}_i) + U(0, \phi_2) \otimes (\mathbf{p}_g - \mathbf{x}_i) + \frac{1}{K_i} \sum_{k=1}^{K_i} U(0, \phi) \otimes (\mathbf{q}_k - \mathbf{x}_i) \right) \quad (2.1)$$

In equation 2.1, the set of K_i neighbors does not include particle i and so, the three parameters ϕ_1 , ϕ_2 , and ϕ are co-existent. Inertial weight ω is kept intact too. It is easy to recognize that equation 2.1 is the general form of velocity update rule. In other words, GPSO is specified by equation 2.1, which balances local best topology and global best topology with expectation that convergence speed is improved but convergence to local optimizer can be avoided. In other words, equation 2.1 aims to achieve both exploration and exploitation. The topology from equation 1.1, equation 1.3, equation 1.4, and equation 1.5 is static (Poli, Kennedy, & Blackwell, 2007, p. 6) because it is kept intact over all iterations of PSO. In other words, neighbors and neighborhood in static topology are established fixedly. However, in the GPSO specified by equation 2.1, it is possible to relocate neighbors of a given particle at each iteration. Therefore, dynamic topology can be achieved in GPSO, which depends on individual applications. This implies that the premature problem can be solved with GPSO so that PSO is not trapped in local optimizer.

In PSO theory, solutions of dynamic problem are to improve the exploitation so that PSO can converge as fast as possible. Inertial weight and constriction coefficient are common solutions for dynamic problem. Hence, GPSO supports tuning coefficients. Concretely, constriction coefficient is tuned with GPSO. However, tuning a parameter does not mean that such parameter is modified simply at each iteration because the modification must be solid and based on valuable knowledge. Fortunately, James Kennedy and Russell C. Eberhart (Poli, Kennedy, & Blackwell, 2007, p. 13), (Pan, Hu, Eberhart, & Chen, 2008, p. 3), (al-Rifaie & Blackwell, 2012, p. 51) proposed bare bones PSO (BBPSO) in which they asserted that, given $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{in})^T$, $\mathbf{p}_i = (p_{i1}, p_{i2}, \dots, p_{in})^T$, and $\mathbf{p}_g = (p_{g1}, p_{g2}, \dots, p_{gn})^T$, the j^{th} element x_{ij} of \mathbf{x}_i follows normal distribution with mean $(p_{ij} + p_{gj})/2$ and variance $(p_{ij} - p_{gj})^2$. Based on this valuable knowledge, I tune constriction parameter χ with normal distribution at each iteration.

Let $\mathbf{z}_i = (z_{i1}, z_{i2}, \dots, z_{in})$ be random vector corresponding to each position \mathbf{x}_i of particle i . Every j^{th} element z_{ij} of \mathbf{z}_i is randomized according to normal distribution $\mathcal{N}\left(\frac{p_{ij} + p_{gj}}{2}, (p_{ij} - p_{gj})^2\right)$ with mean $\mu_i = \frac{p_{ij} + p_{gj}}{2}$ and variance $\sigma_i^2 = (p_{ij} - p_{gj})^2$.

Every z_{ij} is randomized according to normal distribution $\mathcal{N}\left(\frac{p_{ij} + p_{gj}}{2}, (p_{ij} - p_{gj})^2\right)$ (2.2)

Where,

$$\mathcal{N}\left(\mu_i = \frac{p_{ij} + p_{gj}}{2}, \sigma_i^2 = (p_{ij} - p_{gj})^2\right) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{1}{2} \frac{(z_{ij} - \mu_i)^2}{\sigma_i^2}\right)$$

Note, \mathcal{N} denotes normal distribution. Let $g(z_{ij})$ be the pseudo probability density function of z_{ij} .

$$g(z_{ij}) = \exp\left(-\frac{1}{2}\frac{(z_{ij} - \mu_i)^2}{\sigma_i^2}\right) = \exp\left(-\frac{1}{2}\frac{\left(z_{ij} - \frac{p_{ij} + p_{gj}}{2}\right)^2}{(p_{ij} - p_{gj})^2}\right) \quad (2.3)$$

Of course, we have:

$$g(z_{ij}) \sim \mathcal{N}\left(\mu_i = \frac{p_{ij} + p_{gj}}{2}, \sigma_i^2 = (p_{ij} - p_{gj})^2\right)$$

Note, the sign “ \sim ” denotes proportion. Let $X = (\chi_1, \chi_2, \dots, \chi_n)^T$ be *probabilistic constriction coefficient* specified by equation 2.3.

$$\chi_j = \begin{cases} 0 & \text{if } p_{ij} = p_{gj} \text{ and } z_{ij} \neq \frac{p_{ij} + p_{gj}}{2} \\ 1 & \text{if } p_{ij} = p_{gj} \text{ and } z_{ij} = \frac{p_{ij} + p_{gj}}{2} \\ g(z_{ij}) & \text{if } p_{ij} \neq p_{gj} \end{cases} \quad (2.4)$$

Note, X is n -dimension vector. GPSO velocity update rule specified by equation 2.1 is modified as follows:

$$\begin{aligned} \mathbf{v}_i = X \otimes & \left(\omega \mathbf{v}_i + U(0, \phi_1) \otimes (\mathbf{p}_i - \mathbf{x}_i) + U(0, \phi_2) \otimes (\mathbf{p}_g - \mathbf{x}_i) \right. \\ & \left. + \frac{1}{K_i} \sum_{k=1}^{K_i} U(0, \phi) \otimes (\mathbf{q}_k - \mathbf{x}_i) \right) \end{aligned} \quad (2.5)$$

In equation 2.5, constriction coefficient χ is replaced by probabilistic constriction coefficient X . Obviously, equation 2.5 is the most general form of GPSO velocity update rule. According to equation 2.5 with probabilistic constriction coefficient X , the closer to global best position \mathbf{p}_g the local best position \mathbf{p}_i is, the more dynamic the position \mathbf{x}_i is, which aims to exploration for converging to global optimizer. The farer to global best position \mathbf{p}_g the local best position \mathbf{p}_i is, the less dynamic the position \mathbf{x}_i is, which aims to exploitation for fast convergence. This is purpose of adding probabilistic constriction coefficient X to equation 2.1 for solving dynamic problem. As a convention, GPSO specified by equation 2.5 is called *probabilistic GPSO*. Source code of GPSO and probabilistic GPSO is available at https://github.com/locnguyenacademic/sim/tree/master/3_implementation/src/net/ea/psa.

3. Experimental results and discussions

GPSO specified by equation 2.1 and probabilistic GPSO specified by equation 2.5 are tested with basic PSO specified by equation 1.4. The cost function (fitness function) is (Sharma, Chhamunya, Gupta, Sharma, & Bansal, 2015, p. 24):

$$f(\mathbf{x} = (x_1, x_2)^T) = -\cos(x_1)\cos(x_2)\exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2) \quad (3.1)$$

The lower bound and upper bound of positions in initialization stage are $\mathbf{lb} = (-10, -10)^T$ and $\mathbf{ub} = (10, 10)^T$. The terminated condition is that the bias of the current global best value and the previous global best value is less than $\varepsilon = 0.01$. Parameters of GPSO are $\phi_1 = \phi_2 = \phi = 2.05$, $\omega = 1$, and $\chi = 0.7298$. Parameters of probabilistic GPSO are $\phi_1 = \phi_2 = \phi = 2.05$. Parameters of basic PSO are $\phi_1 = \phi_2 = 2.05$ and $\chi = 0.7298$. The swarm size is 50. For the three PSO, dynamic topology is established at each iteration by a so-called fitness distance ratio (FDR). Exactly, Peram (Poli, Kennedy, & Blackwell, 2007, p. 8) defined the topology dynamically at each iteration by FDR. Given target particle i and another particle j , their FDR is the ratio of the difference between $f(\mathbf{x}_i)$ and $f(\mathbf{x}_j)$ to the Euclidean difference between \mathbf{x}_i and \mathbf{x}_j .

$$\text{FDR}(\mathbf{x}_i, \mathbf{x}_j) = \frac{|f(\mathbf{x}_i) - f(\mathbf{x}_j)|}{|\mathbf{x}_i - \mathbf{x}_j|} \quad (3.2)$$

Given target particle i , if $\text{FDR}(\mathbf{x}_i, \mathbf{x}_j)$ is larger than a threshold (> 1), the particle j is a neighbor of the target particle i . Alternately, top K particles whose FDR (s) with \mathbf{x}_i are largest are K neighbors of particle i .

From the experiment, basic PSO, GPSO, and probabilistic GPSO converge to best values -0.9842 , -0.9973 , and -0.9999 with global best positions $(3.0421, 3.1151)^T$, $(3.1837, 3.1352)^T$, and $(3.1464, 3.1485)^T$ after 6, 18, and 18 iterations, respectively. The true best value of the target function specified by equation 3.1 is -1 whereas the true global optimizer is $\mathbf{x}^* = (3.1416, 3.1416)^T$. Therefore, the biases in best values (fitness biases) of basic PSO, GPSO, and probabilistic GPSO are 0.0158, 0.0027, and 0.0001, respectively and the biases in best positions (optimizer biases) of basic PSO, GPSO, and probabilistic GPSO are $(0.0995, 0.0265)^T$, $(0.0421, 0.0064)^T$, and $(0.0048, 0.0069)^T$, respectively.

	Fitness bias	Optimizer bias	Converged iteration
Basic PSO	0.0158	$(0.0995, 0.0265)^T$	6
GPSO	0.0027	$(0.0421, 0.0064)^T$	18
Probabilistic GPSO	0.0001	$(0.0048, 0.0069)^T$	18

Table 3.1. Evaluation of PSO algorithms

From table 3.1, fitness bias and optimizer bias of probabilistic PSO are smallest. Therefore, probabilistic PSO is the preeminent one. Basic PSO converges soonest after 6 iterations but basic PSO copes with the premature problem due to lowest converged fitness value whereas both GPSO and probabilistic GPSO solve the premature problem with better converged fitness values (-0.9973 and -0.9999) but they require more iterations (18). The reason that GPSO is better than basic PSO is combination of local best topology and global best topology in GPSO. The event that probabilistic GPSO is better than GPSO proves that the probabilistic constriction coefficient can solve the dynamic problem.

4. Conclusions

The first purpose of GPSO which is to aggregate important parameters and to generalize important variants is completed with the general form of velocity update rule but the second purpose which is to balance the two PSO properties such as exploration and exploitation is reached at moderate rate although experimental results showed that GPSO and probabilistic GPSO are better than basic PSO due to combination of local best topology and global best topology along with definition of probabilistic constriction coefficient which proved improvement of global convergence. The reason of balance at moderate rate is that dynamic topology in GPSO is supported indirectly via general form of velocity update rule, which is impractical because researchers must modify source code of GPSO in order to define dynamic topology. Moreover, premature problem is solved by many other solutions such as dynamic topology, change of fitness function, adaptation (tuning coefficients, adding particles, removing particles, changing particle properties), and diversity control over iterations. In future trend, I will implement dynamic solutions as much as possible.

References

al-Rifaie, M. M., & Blackwell, T. (2012). Bare Bones Particle Swarms with Jumps. In M. Dorigo, M. Birattari, C. Blum, A. L. Christensen, A. P. Engelbrecht, R. Groß, & T. Stützle (Ed.), *International Conference on Swarm Intelligence. Lecture Notes in*

- Computer Science 7461*, pp. 49-60. Brussels: Springer Berlin. doi:10.1007/978-3-642-32650-9_5
- Pan, F., Hu, X., Eberhart, R., & Chen, Y. (2008, September 21). An Analysis of Bare Bones Particle Swarm. *IEEE Swarm Intelligence Symposium 2008 (SIS 2008)* (pp. 1-5). St. Louis, MO, US: IEEE. doi:10.1109/SIS.2008.4668301
- Poli, R., Kennedy, J., & Blackwell, T. (2007, June). Particle swarm optimization. (M. Dorigo, Ed.) *Swarm Intelligence*, 1(1), 33-57. doi:10.1007/s11721-007-0002-0
- Sharma, K., Chhamunya, V., Gupta, P. C., Sharma, H., & Bansal, J. C. (2015, September). Fitness based Particle Swarm Optimization. (A. K. Verma, P. K. Kapur, & U. Kumar, Eds.) *International Journal of System Assurance Engineering and Management*, 6(3), 319-329. doi:10.1007/s13198-015-0372-4
- Wikipedia. (2017, March 7). *Particle swarm optimization*. (Wikimedia Foundation) Retrieved April 8, 2017, from Wikipedia website: https://en.wikipedia.org/wiki/Particle_swarm_optimization