Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 11, Number 6 (2015), pp. 4997-5018

http://www.ripublication.com

# A Review On Particle Swarm Optimization Algorithm And Its Devolopments

**Dileep M V and Dr. Surekha Kamath**

*Research scholar, Department of Instrumentation and Control Engineering, Manipal Institute of Technology, Manipal University, Manipal Karnataka 576104, India Ph. No: 9895439890 Email. Id: dileeppsla@gmail. com*
*Associate professor, Department of Instrumentation and Control Engineering, Manipal Institute of Technology, Manipal University, Manipal Karnataka 576104, India Email. Id: surekakamath2k4@yahoo. com*

**ABSTRACT**

The optimization is mathematical technique that minimizing or maximizing some parameters of importance from the feasible region. In other words optimization is the selection of a best element on the bunch of alternatives. Particle Swarm Optimization (PSO) is a relatively new, efficient, robust and simple optimization algorithm which proves to work efficiently well on many of these optimization problems. Particle Swarm Optimization is a stochastic multi point search algorithm which models the social behavior of the birds flocking or fish schooling for food. It is widely used to find the global optimum solution in a complex search space. A large number of studies have been done to improve its performance This paper contains the theoretical idea and explanation of the different types of PSO algorithms, selection of the various parameters and their influences, controlling the convergence behaviors of PSO. This paper discussed the advantages and disadvantages of each method tried to highlight them. This paper reviews some kinds of improved versions as well as recent progress in the development of the PSO

**Key words**: Particle swarm optimization, Inertia weight

## 1. INTRODUCTION

The optimization field has an enormous and comprehensive verity of real world applications such as industrial, management etc. Optimization originated in the 1940s, when the British military faced the problem of allocating limited resources (for example fighter airplanes, submarines and so on) to several activities[1]. There is no known single optimization method available for solving all optimization problems. The field of optimization theory is becomes a rapidly growing branch of applied mathematics by the use of modern computers with their high speed computational capabilities. Researchers have applied a qualitatively different approach, using evolutionary algorithms or metaheuristics, to solve parameter optimization problems.

In the last 15-20 years evolutionary and mateheuristics algorithms are developed to a maturity stage. The basic principle used in Evolutionary algorithm is survival of the fittest. Particle swarm optimization is a stochastic population based optimization approach, first published by Kennedy and Eberhart in 1995 [2][3] and its basic idea was originally inspired by simulation of the social behavior of animals such as bird flocking, fish schooling and so on. It is based on the natural process of group communication to share individual knowledge when a group of birds or insects search food or migrate and so forth in a searching space, although all birds or insects do not know where the best position is. But from the nature of the social behavior, if any member can find out a desirable path to go, the rest of the members will follow quickly. Since its first publication, a large body of research has been done to study the performance of PSO, and to improve its performance.

It is a population-based search algorithm and is initialized with a swarm of random particles. PSO makes use of a velocity vector to update the current position of each particle in the swarm. The velocity vector is updated based on the history information gained by the swarm. And the positions of the swarm are updated to search for better positions according to the updated velocity vector. Recently, many researchers have studied the performance of PSO focused mostly on the basic control parameters, namely the acceleration coefficients, inertia weight, velocity clamping and swarm size.

This paper overviews these theoretical studies, and generalizes to more general PSO systems which includes the inertia component. The paper also provides a formal proof that particles converge to a stable point. This point is formally defined. The remainder of the paper is organized as follows: Section 2 provides a short overview of PSO, outlining some of the problems that have been experienced, and proposed solutions. Section 3 overviews the first theoretical study of particle trajectories based on a simplified PSO system. Analysis of constricted trajectories is summarized in Section 4. Trajectory analysis is expanded in Section 5 to include the inertia weight, particles converge to a stable point under specific conditions.

## 2. THE BASIC MODEL OF PSO ALGORITHM

Particle swarm optimization concept was first introduced by Kennedy and Eberhart. They generated the concept of function-optimization by means of a particle swarm. The Particle Swarm Optimization (PSO) algorithm is a multi-agent parallel search technique which maintains a swarm of particles and each particle represents a potential solution in the swarm. All particles fly through a multidimensional search space where each particle is adjusting its position according to its own experience and that of neighbors. Suppose $x_i^t$ denote the position vector of particle $i$ in the multidimensional search space, $R^n$ at time step $t$, then the position of each particle is updated in the search space by,

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (1)$$

where, $v_i^t$ is the velocity vector of particle that drives the optimization process and reflects both the own experience knowledge and the social experience knowledge from the all particles.

Therefore, in a PSO method, all particles are initiated randomly and evaluated to compute fitness together with finding the personal best (best value of each particle) and global best (best value of particle in the entire swarm). After that a loop starts to find an optimum solution. In the loop, first the particles' velocity is updated by the personal and global bests, and then each particle's position is updated by the current velocity. The loop is ended with a stopping criterion predetermined in advance. Basically, two PSO algorithms, namely the Global Best ($G_{best}$) and Local Best ($l_{best}$) PSO, have been developed which differ in the size of their neighborhoods.

Mainly PSO algorithms are of two types based on the size of their neighborhoods particles. Global best PSO and Local best PSO. In global best PSO the position of each particle is influenced by the particle which is having best-fitness value in the entire swarm. In this method it is possible to identify the social behavior of all particle s in the entire swarm. Like other population-based optimization techniques, convergence speed and global search ability are the two critical criteria for evaluating the performance of PSO algorithms. In the original PSO [2][3][4][5], all particles learn from $G_{best}$ in updating velocities and positions. Hence the algorithm exhibits a fast-converging behavior.

Let as consider particles $i \in [1, \dots\dots, n]$ where $n > 1$, has a current velocity and position $x_i, v_i$ respectivily. The personal best position $P_{best,i}$ ie, the position in the search space where particle i had the optimum value as determined by the fitness function f. In this method the personal best is called the global best. For a minimization problem the personal best is updated using the following equations.

$$P_{best,i}^{t+1} = \begin{Bmatrix} P_{best,i}^t \; if \; f(x_i^{t+1}) > P_{best,i}^t \\ x_i^{t+1} \; if \; f(x_i^{t+1}) \leq P_{best,i}^t \end{Bmatrix} \quad (2)$$

where $t$ is the time step

$G_{best} = min\{P_{best,i}^t\}$, *where* $i \in [1, \dots\dots, n]$ *and* $n > 1$

In this the global best position is the best position discovered by any of the particles in the entire swarm. The velocity of the particle i is calculated by

$$v_{ij}^{t+1} = v_{ij}^t + c_1 r_{1j}^t [P_{best,i}^t - x_{ij}^t] + c_2 r_{2j}^t [G_{best,i} - x_{ij}^t] \quad (3)$$

$c_1 and c_2$ are positive acceleration constants which are used to level the contribution of the cognitive and social components respectively $r_{1j}^t$ and $r_{2j}^t$ are random numbers from uniform distribution $U(0, 1)$ at time t.

The local best PSO method only allows each particle to be influenced by the best-fit particle chosen from its neighborhood, and it reflects a ring social topology. Here this social information exchanged within the neighborhood of the particle, denoting local knowledge of the environment and the velocity is calculated by,

$$v_{ij}^{t+1} = v_{ij}^t + c_1 r_{1j}^t [P_{best,i}^t - x_{ij}^t] + c_2 r_{2j}^t [l_{best,i} - x_{ij}^t] \quad (4)$$

where, $l_{best,i}$ is the best position that any particle has had in the neighborhood of particle found from initialization through time t.
In PSO, particle velocity is very important, since it is the step size of the swarm. At each step, all particles proceed by adjusting the velocity that each particle moves in every dimension of the search space. Mainly the search characteristics is of two type, exploration and exploitation[7]. These two characteristics have to balance in a good optimization algorithm[6]. Parameter choice in PSO will results some problems in original version of PSO. They are explained below.

### 2. 1 Velocity climping

In velocity update equation it is having two components. First is called cognitive component, $c_1 r_{1j}^t [P_{best,i}^t - x_i^t]$ which measures the performance of the particles relative to its past experience. Second is called the social component, $c_2 r_{2j}^t [lG_{best,i} - x_i^t]$ which measures the performance of the particles relative to a group of particles or neighbors. Due to the high values of acceleration coefficients the velocity of the particles quickly rises to large values, especially for particles far from their $G_{best}$ and $P_{best}$positions. Which results the particles are having large position updates with particles leaving the boundaries of their search space. To control the increase in velocity, velocities are clamped[8]
Now if a particle's velocity goes beyond its specified maximum velocity $\mathrm{v_{max}}$, this velocity is set to the value $\mathrm{v_{max}}$and then adjusted before the position update by,

$$v_i^{t+1} = min\left(v'^{t+1}_i, v_{max}\right) \tag{5}$$

Where $v'^{t+1}_i$ is the velocity update.
If the maximum velocity $v_{max}$is too large, then the particles may move erratically and jump over the optimal solution. On the other hand, if $v_{max}$is too small, the particle's movement is limited and the swarm may not explore sufficiently or the swarm may become trapped in a local optimum. This problem can be solved by calculate $\mathrm{v_{max}}$ as a fraction of the domain search space.

$$v_{max} = \varphi(x_{max} - x_{min}) \tag{6}$$

Where $x_{max}$ and $x_{min}$ are the upper and lower bound of the particles. $\varphi$ is the random number, $\varphi \epsilon (0,1]$
In this method if all particle's velocities are equal to $v_{max}$ the large velocity updates will occur. To solve this problem $v_{max}$ can be reduced over time. The advantage of velocity clamping is that it controls the explosion of velocity in the searching space. Andries Engelbrecht[9] answered for the question, how velocities should be initialized. Though zero initial velocities have been advocated, a popular initialization strategy is to set initial weights to random values within the domain of the optimization problem.

## 2. 2 Inertia weight

Inertia weight is used to replace the role of $v_{max}$ by adjusting the influence of the previous velocities. It is done by controlling the momentum of the particle by weighing the contribution of the previous velocity. So the velocity update equation will change as,

$$v_{ij}^{t+1} = \omega v_{ij}^{t} + c_1 r_{1j}^{t}[P_{best,i}^{t} - x_{ij}^{t}] + c_2 r_{2j}^{t}[G_{best,i} - x_{ij}^{t}] \quad (7)$$

Where $\omega$ is called the inertia weight.

If $\omega \geq 1$ then the velocities increase over time and particles can hardly change their direction to move back towards optimum, and the swarm diverges. If$\omega \ll 1$ then little momentum is only saved from the previous step and quick changes of direction are to set in the process. If $\omega = 1,$ particles velocity vanishes and all particles move without knowledge of the previous velocity in each step. Initially the inertia value used to be fixed value for every updates, but now it is changing dynamically to control exploration and exploitation of the search space. Usually the large inertia value is high at first, which allows all particles to move freely in the search space at the initial steps and decreases over time. To optimize the global and local exploration rate the inertia weight should decrease linearly with the iteration and it is denoted as,

$$\omega^{t+1} = \omega_{max} - \left(\frac{\omega_{max} - \omega_{min}}{t_{max}}\right) t \quad (8)$$

Where t is the iteration time.

Van den Bergh and Engelbrecht[10] have defined a condition which guarantees convergence

$$\omega > \frac{1}{2}(c_1 - c_2) - 1 \quad (9)$$

For ordinary cases the inertia weight variation is linear from *0. 9 to. 04* over the iteration

The inertia weight technique is very useful to ensure convergence. This is having a disadvantage is that once the inertia weight is decreased, it cannot increase if the swarm needs to search new areas. This will cause the premature convergence.

## 2. 3 Acceleration coefficients

The stochastic influence of the cognitive and social components of the particle's velocity is controlled by the random values $r_1$ *and* $r_2$ together with the acceleration coefficients. The cognitive component gives the confidence level of the individual particles and the social component gives the confidence level of the neighboring particles.

When $c_1 > 0$ *and*$c_2 = 0$, all particles are independent. The velocity update equation will be,

$$v_{ij}^{t+1} = v_{ij}^{t} + c_1 r_{1j}^{t}[P_{best,i}^{t} - x_i^{t}] \quad (10)$$

When $c_1 = 0$ *and* $c_2 > 1$, all particles are attracted to a single point in the entire swarm and the update velocity will become,

$v_{ij}^{t+1} = v_{ij}^{t} + c_2 r_{2j}^{t}[G_{best,i} - x_i^{t}]$for gbest PSO,

$v_{ij}^{t+1} = v_{ij}^{t} + c_2 r_{2j}^{t}[l_{best,i} - x_i^{t}]$for lbest PSO, When $c_1 = c_2$, the particles are having same effect on both $P^t_{best,i}$ *and* $G_{best}$

When $c_1 \gg c_2$, each particle is more strongly influenced by its personal best position, resulting in excessive wandering. In contrast, when $c_2 \gg c_1$ then all particles are much more influenced by the global best position, which causes all particles to run prematurely to the optima[11].

### 2. 4 Constriction coefficient

The term constriction coefficient was introduced by Clerc[6][12]. This is the scaling parameter which plays an important role in exploration and exploitation mechanism of the particles to ensure convergence behavior. Clerc chose to use an alternate representation for the velocity update equations.

$$v^{t+1}_{ij} = X\{v^t_{ij} + \emptyset_1[P^t_{best,i} - x^t_i] + \emptyset_2[l_{best,i} - x^t_i]\} \quad (11)$$

Where, $X = \frac{2}{2-\emptyset-\sqrt{\emptyset^2-4\emptyset}}, \emptyset = \emptyset_1 + \emptyset_2, \emptyset_1 = c_1r_1 \text{ and } \emptyset_2 = c_2r_2$

According to Bratton[13] if, $\emptyset < 4$then all particles would slowly spiral toward and around the best solution in the searching space without convergence guarantee. If, $\emptyset > 4$, then all particles converge quickly and guaranteed. By the use of the constriction coefficient, the amplitude of the particle's oscillation will be decreased and it will focuses on the local and neighborhood previous best points. The particle will perform local search if the neighborhood best position and previous best positions are near to each other and particles will perform global search if they are far from each other. The particle may follow on a cyclic path like an oscillation because of the large separation between personal best position and the neighborhood best position. This will cause the non-convergence of the solutions and it the drawback of constriction coefficient[14].

## 3. GUARANTEED CONVERGENCE PSO (GCPSO)

If a particle's current position coincides with the global best position, the particle will only move away from this point if its previous velocity and $\omega$ are non-zero. in other words, when$x^t_{ij} = p^t_{ij} = G^t_{best}$, then velocity update will only depend upon the first term in velocity update equation ie, $\omega v^t_{ij}$. If their previous velocities are very close to zero, then all the particles will stop moving once they catch up with the global best particle, which may lead to premature convergence of the algorithm. This does not even guarantee that the process has converged to a local minimum, it only means that all particles have converged to the best position in the entire swarm[11][15][16]. This leads to stagnation of the search process which the PSO algorithm can overcome by forcing the global best position to change when$x^t_{ij} = p^t_{ij} = G^t_{best}$. To solve this particular problem a new parameter is introduced to the PSO. Let τ be the index of the global best particle, so that

$$y_\tau = G_{best} \quad (12)$$

A new velocity updates equation for the globally best positioned particle, $y_\tau$ has been suggested,

$$v_{\tau j}^{t+1} = -x_{\tau j}^{t} + G_{best}^{t} + \omega v_{\tau j}^{t} + \rho^{\tau}\left(1 - 2r_{2j}^{t}\right) \quad (13)$$

Where, $'\rho^{\tau}'$ is a scaling factor and causes the PSO to perform a random search in an area surrounding the global best position $G_{best}$.

The above equation the last term$\rho^{\tau}\left(1 - 2r_{2j}^{t}\right)$generates a random sample from a sample space with side lengths $2\rho^{\tau}$[11]. The position updating equation will become,

$$x_{\tau j}^{t+1} = G_{best}^{t} + \omega v_{\tau j}^{t} + \rho^{\tau}\left(1 - 2r_{2j}^{t}\right) \quad (14)$$

While all other particles in the swarm continue using the usual velocity update equation and the position update equations. The parameter $\rho^{\tau}$controls the diameter of the search space.

$$\rho^{\tau+1} = \begin{cases} 2\rho^{\tau} \; if \; successes(t) > \epsilon_s \\ 0.5\rho^{\tau} \; if \; failures(t) > \epsilon_f \\ \rho^{\tau} \; otherwise \end{cases} \quad (15)$$

Where $\#successes$ and $\#failures$ respectively denote the number of consecutive successes and failures, and a failure is defined as[17],

$$f(G_{best}^{t+1}) = f(G_{best}^{t}) \quad (16)$$

The following conditions must also be implemented to ensure that the above equation is well defined.

$$\left.\begin{matrix} \#successes(t+1) > \#successes(t) \rightarrow \#failures(t+1) = 0 \\ \#failures(t+1) > \#failures(t) \rightarrow \#successes(t+1) = 0 \end{matrix}\right\} \quad (17)$$

Therefore, when a success occurs, the failure count is set to zero and similarly when a failure occurs, then the success count is reset. The optimal choice of values for$\epsilon_s$ *and* $\epsilon_t$depend on the objective function. It is difficult to get better results using a random search in only a few iterations for high- dimensional search spaces, and it is recommended to use $\epsilon_s = 15$ and $\epsilon_f = 5$. On the other hand, the optimal values for $\epsilon_s$ and $\epsilon_t$ can be found dynamically. For instance, $\epsilon_s$ may be increased every time that$\#failures(t) > \epsilon_f$ *i. e.* it becomes more difficult to get the success if failures occur frequently which prevents the value of ρ from fluctuating rapidly. Such strategy can be used also for $\epsilon_f$.

## 4. RECENT WORKS AND ADVANCED TOPICS

### 4. 1 Multi-Start PSO (MSPSO)

One of the major problems with the basic PSO is lack of diversity when particles start to converge to the same point. Several models have introduced to reduce this particular issue by continually injecting randomness, or chaos, into the swarm[7][18]. These types of methods are called the Multi-start (or restart) Particle Swarm Optimizer (MSPSO). This method attempts to detect when the PSO has found a local minimum. Once a local minimum is found, the algorithm re-starts the algorithm with new randomly chosen initial positions for the particles. [11].

The advantages of using randomly reinitializing particles, is first mentioned by Kennedy and Eberhart[2]. The particular process is referred ascraziness. Although Kennedy mentioned the potential advantages of a craziness operator, no evaluation of such operators was given. When considering any method to add randomness to the swarm, a number of aspects need to be considered, including what should be randomized, when should randomization occur, how should it be done, and which members of the swarm will be affected. The diversity of the swarm can be increased by randomly initializing position and velocity[19][20]. When position vectors are kept constant and velocity vectors are randomized, particles preserve their memory of current and previous best solutions, but are forced to search in different random directions. Let $c_v$ *and* $c_p$ are the chaos factors for velocity and position respectively. *If* $r_{ij} \sim U(0,1) < c_v$ then the particle velocity component is reinitialized to $v_{ij}^{t+1} = U(0,1)V_{max,j}$ where $r_{ij}$ is random number for each particle $i$ and each dimension $j$. Again, *if* $r_{ij} \sim U(0,1) < c_p$ then the particle position component is initialized to $x_{ij}^{t+1} \sim U(x_{min,j}, x_{max,j})$. The high chaos factor values in the beginning and will decrease over time to ensure an equilibrium state can be reached.

Massimiliano Kaucic et. al[21] carried out a study in adaptive velocity based on the differential operator enhances the optimization ability of the particles. The main novelty of the procedure is the integration of the opposition-based computing into a PSO with an adaptive velocity used in order to produce some additional exploration ability of the search space. The strategy uses the super-opposition paradigm to re-initialize particles in the swarm. Xiao-Feng Xie et. al[22] introduce the adaptive criterion is appended on individual level. By analyzing the social model of PSO, a replacing criterion based on the diversity of fitness between current particle and the best historical experience is introduced to maintain the social attribution of swarm adaptively by taking off inactive particles.

An improved PSO (IPSO) algorithm is introduced by Ziyang Zhen et. al[23], in which each particle has the ability of keeping itsinertia motion and learning from another randomly selected particle with better performance; moreover, for the particlewith better performance, the inertia weight will be larger and the learning coefficient will be smaller. Thus, for the particles sorted in order of decreasing performance, the inertia weights are decreased and the learning rate coefficients are increased gradually. Similar studies were carried out in with two learning factors and the inertia weight[24] and dynamically changing inertia weight[25].

### 4. 2 Multi-phase PSO (MPPSO)

Multi-phase PSO (MPPSO) method partitions the main swarm of particles into sub-swarms or subgroups. In this different groups of particles have trajectories that proceed along trajectories with differing goals in different phases of the algorithm. B. Al-Kazemi and C. Mohan [26][27][28] introduced this method by dividing the main swarm into two sub swarms of equal size. Particles are randomly assigned in this algorithm. There are two phases in the algorithms, attraction and repulsion phase.

Particles of the sub swarms are influenced to move towards the global best position in attraction phase and go away in repulsion phase. Sub-swarms switch phases either when the number of iterations in the current phase exceeds a user specified threshold, orwhen particles in any phase show no improvement in fitness during a user specified number of consecutive iterations[7].

In MPPSO algorithm the velocity updating equation is given by,

$$v_{ij}^{t+1} = \omega v_{ij}^{t} + c_1 x_{ij}^{t} + c_2 G_{best} \tag{18}$$

In the above equation the personal best term is eliminated, since a particle's position is only updated when the new position improves the performance in the solution space[29]. The advantage of the MPPSO algorithm is that when the fitness of a particle doesn't changed any more, then the particle's flying speed and direction in the searching space are changed by the adaptive velocity strategy. Multi- swarm[30], consepts are also introduced in MPPSO algorithm Zhiqiang Genget. al[31] divided the whole swarm into three sub-swarms randomly in his studies.

## 4. 3 Random Particle Approach (RPSO)

The simplest way to construct a PSO-based global search algorithm is to introduce randomized particles to the swarm. Particle can be made a randomised particle by simply resetting its position to a random position in search space periodically. Any of the particles in the swarm can be made as random particles, but the size of the swarm will depends on the optimal ratio of random versus normal particles. In this algorithm the position of the particular particle will change in every iteration, allowing the particle to explore the region in which it was initialized before resetting it again. The resulting algorithm is called the Randomized Particle Swarm Optimizer, or RPSO[11].

## 4. 4 Perturbed PSO (PPSO)

In classical PSO method rate of the swarm convergence is high with in the intermediate vicinity of the $G_{best}$. This results the lost of diversity and premature convergence if the $G_{best}$ corresponding to the local minima. To overcome this two issues a perturbed particle swarm algorithm which is based upon a new particle updating strategy and the concept of perturbed global best ($p - G_{best}$) within the swarm is introduced by Zhao Xinchao et. al[32]. The perturbed global best (*p-gbest)* updating strategy is based on the concept of possibility measure to model the lack of information about the true optimality of the gbest. In velocity update equation the $G_{best}$ is replaced by $G'_{best}$ and is the normal distribution function of $G_{best}$ and $\sigma$. Where $\sigma$ represents the degree of uncertainty about the optimality of the $G_{best}$.

$$v_{ij}^{t+1} = \omega v_{ij}^{t} + c_1 r_{1j}^{t}\left[P_{best,i}^{t} - x_{ij}^{t}\right] + c_2 r_{2j}^{t}\left[G'_{best} - x_{ij}^{t}\right] \tag{19}$$

where, $G'_{best} = N(G_{best,\sigma})$is the $j^{th}$ dimension of $p - G_{best})$ int$^{th}$ iteration.

$\sigma$simply expressed as,

$$\sigma = \begin{Bmatrix} \sigma_{max} \; iterations < \alpha \times max_{iterations} \\ \sigma_{min} \; otherwise \end{Bmatrix} \tag{20}$$

Where $\sigma_{max}$ *and* $\sigma_{min}$ and $\alpha$ are manually set parameters[17].

$N(G_{best,\sigma})$ is the normal distribution, and represents the degree of uncertainty about the optimality of the $G_{best}$.

The perturbed global best updating strategy is different from conventional mutation operator which applies a random perturbation to the particles. The function of $(p - G_{best})$ is to encourage the particles to explore a region beyond that defined by the search trajectory. When$\sigma$ is large and encourages local fine-tuning at the latter stage when $\sigma$ is small. Subsequently, this approach helps to reduce the likelihood of premature convergence and guides the search toward the promising search area[32]. Abdelaziz Laifaet et. al[33] introduced a linear model to $\sigma$ expression.

$$\sigma = \sigma_{max} - \frac{iteration - 1}{\underset{iterations}{max}}(\sigma_{max} - \sigma_{min}) \tag{21}$$

The study has been done in Stochastic Perturbing Particle Swarm Optimization byMehmet Sevkli and Aise Zulal Sevkli [34]. In this the position of each particle is updated based on the personal best and the global best. These operations in Stochastically Perturbed-PSO(SPPSO) are similar to classical PSO algorithm but, the search strategy of particles is different. Insert function $(\eta\ )$ is introduced to improve the randomness. Lei Zhang et. al[35] carried out the SPPSO study by considering more probability variables like Perturbation Probability, Gaussian noise, Acceptance Probability. Function reduces the likelihood of premature convergence and also helps to direct the search towards the minima. Fulong Chen et. al[36] explained the new approach called partially Perturbed Particle Swarm Optimization. The algorithm enhances the capability of conventional particle swarm optimization (CPSO) by partially perturbing the coordinates of the globally best particle with the patterns when the searching process is locally confined.

### 4. 5 Multi-Objective PSO (MOPSO)

The process of optimizing systematically and simultaneously a collection of objective functions are called multi-objective optimization (MOO) or vector optimization[37]. MOPSO was proposed by Moore et. al[38], to optimize more than one objective functions. In case of a multi-objective optimization problem, there exists no single definition of the optimal solution. Basically, the problem is that, how the individual objective functions should be weighted in relation to each other. But in the single objective optimization there is no such problem, since there is only one objective[39]. The main characteristics of multi-objective optimization are the objective function weight allocation. The solution for the weighting problem is a natural basis for the classification MOO[40]. There exist a set of solutions for the multi-objective problem which cannot normally be compared with each other. Such solutions are called Pareto optimal solutions or non-dominated solutions.

There are two main fundamental approaches in the MOPSO algorithm[41][42]. The first approach is that each particle will consider a single objective function at a time. In this the best position of the particle is finding is similar to the single-objective optimization case and the main challenge is the proper management of the information coming from each objective function so that the particles go toward Pareto optimal solutions. The second approach is that each particle will consider all objective functions and based on the concept of Pareto optimality, and they produce non-dominated best positions called leaders[45][43][44][47][46]. The selection of the best position is an important task in making particles move to the Pareto optimal front[48]. In these approaches, to find the leaders is not straight forward task, since there may be many non-dominated solutions in the neighborhood of a particle, but only one is usually selected to participate in the velocity update.

Carlos et. al[42] introduce special mutation operator to enhance the exploratory capabilities of MOPSO and uses a secondary repository of particles used by other particles to guide their own flight. The review of contrained MOPSO has done by Nor Azlina et. al[49]. Yang et. al[50] propose two feature selection algorithms, which are based on two inertia weight strategies to properly balance the local search and global search of PSO. Bing Xue et. al[51] presents the study on multi-objective particle swarm optimization (PSO) for feature selection. They also developed an algorithm which applies the ideas of crowding, mutation, and dominance to PSO to search for the Pareto front solutions.

### 4. 6 Dynamic Neighborhood PSO (DNPSO)

The multiple objectives are divided into two groups: $f_1$ and $f_2$, where $f_1$ is defined as the neighborhood objective, and $f_2$ is defined as the optimization objective. The choices of $f_1$ and $f_2$ are arbitrary. In each iteration, each particle dynamically determines a new neighborhood by calculating the distance to all other particles and choosing the $m$ nearest neighbors[52]. The distance is described as the difference between fitness values for the first group of objective functions $f_1$. When the neighborhood has been determined, the best local value is selected among the neighbors in terms of the fitness value of the second objective functions $f_2$. Finally, the global best updating system considers only the solution that dominates the current personal best value[14].

In each generation, after calculating distances to every other particle, each particle finds its new neighbors. Among the new neighbors, each particle finds the local best particle as the $l_{best}$[53][54]. Burak et. al[55] proposed three different methods for determining the particle neighbors, which are namely nearest neighbors in the search space and function space[56], random neighborhood and Directed Graph Representation of the Neighborhood. Ehsan et. al[57] introduced a naturalistic method to determine neighbors in a set of particles based on Voronoi diagram. In this realistic swarm, particles take Voronoi neighbors into account. The study which incorporate PSO and the neighborhood rough set model is introduced by Aimin et. al[58]. Neighborhood rough set model is difinied by neighborhood and neighborhood relations in metric spaces. In this model, the neighborhood of an arbitrary $x_i$ is a subset of samples, whose attribute to describe the objects is close to $x_i$. Min Han et. al[59] carried out a study in DynamicNeighborhood Topology for Large Scale Optimization. In this the adaptive topology strategy could learn from the neighborhood positions, and constitute the sub-swarm self-organized to optimize the issue and share the information. Some of the studied was carried out in the con trained DNPSO[60][61]. The main drawback o this algorithm is that it can be used only for two objectives[62].

### 4. 7 Vector Evaluated PSO (VEPSO)

Vector evaluated PSO is similar to the multi-phase PSO. Here one swarm is evaluated only for one objective function and the information is exchanged among other swarms[63]. As a result in the swarm update, the best position of one swarm is used for the velocity update of another swarm that corresponds to a different objective function. Which utilizes a separate Swarm for each optimization objective and communication between the swarms to encourage exploration of the Pareto front[64][65]. Thus the information exchange will took place between swarms and the Pareto optima can be find out. In this algorithm the number of objectives is equal to the number of swarms. In other words $k$ objective functions are solved by $k$ swarms. The velocity update equation for k-objective function problem is defined as,

$$v_{ij}^{[s]} = \omega v_{ij}^{[s]} + c_1 r_1 \left[p_{best,i}^{[s]}(t) - x_{ij}^{[s]}(t)\right] + c_2 r_2 \left[G_{best}^{[q]}(t) - x_{ij}^{[s]}(t)\right] \qquad (22)$$

Where, $s$ defines the swarm number

$G_{best}^{[q]}$ is the best position found for any particle in the $q$ −th swarm which is evaluated with the $k$-th objective function.

Zuwairie Ibrahim*et. al*[66] is solved the DNA sequence design problem is by using Vector evaluated-PSO(VEPSO). In this they considered four objective functions to minimize. A study is done by Lim Kian Shenget *et. al*[67] to measure the convergence and diversity by using standard test functions.

### 4. 8 Binary PSO (BPSO)

The classical PSO was developed for continues-valued real search space and its variants are also most dealing with the continuous space. Which cannot be used to for a discrete valued search space [74]. Kennedy and Eberhart first introduced the basic BPSO version which can be used in the discrete space. It is possible to convert the real value domain to binary value domain and wise versa. This is can be found out in analog to digital conversion. So Kennedy and Eberhart are developed the PSO to operate on the binary search spaces. In this algorithm the particle position are represented by the binary value $0 \ or \ 1$[68][69].

$$x_{ij} \in \{0, 1\} \qquad (23)$$

In this, the mapping can be done from the n-dimensional binary space $B^n$ to the real number$f: B^n \rightarrow R$.

Where $f$ is the function and $R$ is the real number set. $n$is bit string length.

That means a particle's positions must belong to $B^n$ in order to be calculated by$f$. In this algorithm the particle velocity update equation is not having any change. The particle update is taking place using the following formula. For the $j^{th}$ bit of the $i^{th}$particle, the next update is,

$$x_{ij}^{t+1} = \begin{cases} 1 \ if u_{ij}^t < s_{ij}^t \\ 0 \ if u_{ij}^t > s_{ij}^t \end{cases} \qquad (24)$$

where, $u_{ij}^t$ is a random number in the interval (0, 1), $s_{ij}^t$ is the sigmoid function.

Since each bit of $x_{ij}$is always binary-valued in the solution space, so no boundary conditions need to be specified in BPSO[70]. The probability based Binary Particle Swarm Optimization Algorithm is developed by LanLan et. al[71][72] they transform the pseudo –probability of the particle $x_{ij}$to the real probability $p_{ij}$.

$$p_{ij} = \frac{x_{ij} - x_{min}}{x_{max} - x_{min}} \quad (25)$$

Arezoo Modiri et. al[73]modified the algorithm to improve the convergence. They studied the impact of modifying the initial velocity and the effect of omitting personal best coefficient in velocity function on performance of PSO algorithm.

## 4. 9 Theta PSO ($\theta - PSO$)

The theta-PSO algorithm is a updated version of classical PSO[75][76][77][78][79]. The theta-PSO is having very similar structure that of classical PSO. This method is having different moving mechanism of particles. Phase angle vector is update can done instead of velocity vector update classical PSO. Similar operations are carrying outin θ-PSO, the increment of the phase angle vector $\Delta\theta_i$is used instead of velocity vector, in each iteration the particle's phase angles $\theta_i$ is updating. The positions of particles can be found out by the mapping of phase angles using an appropriate functions.

$$\Delta\theta_i^{t+1}(d) = w * \Delta\theta_i^t(d) + c_1 * rand\left(\theta p_i^t(d) - \theta_i^t(d)\right)$$
$$+c_2 * rand\left(\theta p_g^t(d) - \theta_i^t(d)\right) \quad (26)$$
$$\theta_{id}(t+1) = \theta_{id}(t) + \Delta\theta_{id}(t+1) \quad (27)$$
$$X_{id}(t) = f\left(\theta_{id}(t)\right) \quad (28)$$

$F'_i(t) =$ Fitness value $X_i(t)$

With $\theta_{id} \in (\theta_{min}, \theta_{max})$, $\Delta\theta_{id} \in (\Delta\theta_{min}, \Delta\theta_{max}), X_{id} \in (X_{min}, X_{max})$being a monotonic mapping function for common problems we take$\theta_{id} \in \left(-\pi/_2, \pi/_2\right)$, $\Delta\theta_{id} \in \left(-\pi/_2, \pi/_2\right)$

$$f(\theta_{id}) = \frac{x_{max} - x_{min}}{2} sin\theta_{id} + \frac{x_{max} - x_{min}}{2} \quad (29)$$

Where $d = 1,2, \ldots, D, i = 1,2, \ldots, S$

An study the fuzzy adaptive theta particle swarm optimization (FAΘPSO) is carried out by Siavash[80]. They try to combine the fuzzy as well as theta particle swarm optimization.

## 4. 10 Cooperative PSO-Split (CPSO-S)

The particular algorithm was introduced by van denet et. al[81][82]. In CPSO the entire search space is splitting into subspaces it similar to Multi-swarm PSO, where the total swarms are divided into sub swarms[83]. CPSO-S was developed as an approach to improve the scalability of the PSO. CPSO-S moves individual components in beneficial directions without affecting the rest of the particles. It is done byconverting the n-dimensional search space into $n - 1$ dimensional spaces, each of which is optimized independently by a subswarm. Each subswarm is a

complete implementation of a PSO variant. Each swarm in the group only has information regarding a specific component of the solution vector; the rest of the vector is provided by the other n − 1 swarms. This promotes cooperation between the different swarms. The information based on specific components of the solution vector is carried out by each group of swarms. $n-1$swarms providing the rest of the vectors. This particular character provides different swarms cooperation.
In this algorithm the working of each sub-swarm are relatively independent in nature. Subswarms are doing the cooperative search between them and will try to achieve the global minimapoint. The particle swarm coordinator will periodically send the information. Particle swarm coordinator first receives global extreme particles $p_{Gbest-1}$ from sub-PSO which is received from coordinated cycle. Then this global extreme particle is coordinated by $t$ particle swarm coordinator and the best particle will be selected according to the fitness value. Thus the global extreme particle $p_{Gbest}$of all particles can be found out. Distribution of global extreme particle $p_{Gbest}$can be done in various sub-swarms. Finally global best particle replacement will be done based on their complete strategy[84][85]..
The CPSO-S approach is having the advantage that modification of a single component can be done at a time. This results the required fine grained search and it will completely reduce the scenarios like "two steps forward, one step back". Large number of combinations will form in CPSO-S using different swarm members, this will result to the significant increase in the solution diversity[86]. Fuqiang Xie et. al[87] introduced a Cooperate PSO based on Legendre orthogonal polynomials. They introduced three improving strategies which ensure the better in solution in complex space.

### 4. 11 Quantum Behaved Particle Swarm Optimization (QPSO)

In classical PSO the position of the particle is updated by the velocity. But in quantum mechanics it is not possible to find a particle position and velocity simultaneously. In QPSO algorithm[88] used the quantum behavior of the particle. It says that in each dimension there is a one dimensional potential well on it witha local attractor point. Instead of using position and velocity the quantum state of a particle is used by a wave function[89][90]. From Schrodinger equation it is possible to find the probability density function of the particular particle.
In quantum time-space, the particle is having both particle and wave nature[91], the wave function $|\Psi(x,t)|^2$can explain quantum state of a particle. Therefore the wave function $|\Psi(x,t)|^2$gives the probability of the particle's appearing in position x. The particle position can be find out by using Monte Carlo simulation.

$$x(t) = p \pm 0.5L \times ln(1/_u) \quad (30)$$

Where $u \in [0,1]$ and $L$ is defined as,

$$L(t+1) = 2\beta \times |mbest - x(t)| \quad (31)$$

Where, $\beta$and $mbest$are called creativity coefficient and mean best position respectively. The term $mbest$is defined as,

$$mbest = \frac{1}{m}\sum_{1=1}^{m} p_i \quad (32)$$

where d is particle's dimension, $m$is size of the population, $p_i$is the particle's best position. The centre potential is calculated to ensure global convergence to all particles and it is defined as,

$$p = \varphi \times p_{best} + (1-\varphi)G_{best} \quad (33)$$

Where, $\varphi \in [0,1]$ is a random number. $G_{best}$ is the global best position of the particle and $p_{best}$personal best position of particle. Therefore, the particle position update equation is given by,

$$x(t+1) = p \pm \beta \times |mbest - x(t)| \times ln(^1/_u) \quad (34)$$

In general, in the particle update equation is controlled by the parameter $\beta$ and it is expresses as,

$$\beta = \beta_{min} + (\beta_{max} - \beta_{min})\frac{T-t}{T} \quad (35)$$

Where, T and t represents the maximum number of iterations and the current iteration respectively. The quantum-behaved particle swarm Optimization(QPSO) algorithm is easily controllable because it is using a single parameter $\beta$ to control the position of the particle as well as convergence rate.

## 5. CONCLUTION

Evolution of Optimization techniques has a very long history and still new techniques are developing day by day. This paper attempts to comprehend the evolution of optimization techniques and the modification made on each during various periods. By the development of engineering, application of optimization have varied from a simple process to complex process like missile trajectory optimization and so on. So as the development of technology new algorithms, ideas and techniques have to be developed. In the literatures reviewed, we came to a conclusion that even if the technique of optimization to be used, largely depends on the application, co-operative and evolutionary methods have more importance due to their effectiveness in solving particular problems and due to the ease of implementation. Recently developed quantum particle swarm optimization technique is giving new promises in the area of optimization. Also the modification of old methods is also going on. The research on optimization is a never ending process. No literature review on this topic will be complete since new algorithms are developing each day. This paper is just a small effort to showcase what have been done till now.

## REFERENCES

[1] Singiresu S. Rao, Engineering Optimization Theory and Practice, 4th edition, Ed.: John Wiley and Sons, 2009.

[2] J. Kennedy and R. C. Eberhart, Particle Swarm Optimization, Proceedings of the IEEE International Conference on Neural Networks, Vol. 4, pp. 1942-1948, 1995

[3] Eberhart, R. C. and Kennedy, J. A new optimizer using particle swarm theory. Proceedings of the Sixth International Symposium on Micromachine and HumanScience, Nagoya, Japan. pp. 39-43, 1995

[4] Ioan Cristian Trelea “The particle swarm optimization algorithm: convergence analysis and parameter selection” Information Processing Letters 85 (2003) 317–325

[5] X. Li and K. Deb, "Comparing lbest PSO Niching algorithms Using Different Position Update Rules, " in In Proceedings of the IEEE World Congress on Computational Intelligence, Spain, 2010, pp. 1564-1571.

[6] M. Clerc, J. Kennedy, The particle swarm: explosion stability and convergence in a multi-dimensional complex space, IEEE Trans. Evolution. Comput. 6 (1) (2002) 58–73.

[7] Andries P. Engelbrecht, Computational Intelligence: An Introduction.: John Wiley and Sons, 2007, ch. 16, pp. 289-358.

[8] R. C. Eberhart, P. K. Simpson, R. W. Dobbins, Computational Intelligence PC Tools, first ed., Academic Press Professional, 1996.

[9] Andries Engelbrecht “Particle Swarm Optimization: Velocity Initialization” WCCI 2012 IEEE World Congress on Computational Intelligence June, 10-15, 2012

[10] Peer, E. S. ; van den Bergh, F. ; Engelbrecht, A. P “Using neighbourhoods with the guaranteed convergencePSO” Swarm Intelligence Symposium, IEEE 2003.

[11] F. van den bergh, An Analysis of Perticle Swarm Optimizers. PhD thesis, Department of Computer Science., 2006, University of Pretoria, Pretoria, South Africa.

[12] M. Clerc, The swarm and the queen: towards a deterministic and adaptive particle swarm optimization, in: Proc. ICEC, Washington, DC, 1999, pp. 1951–1957.

[13] Bratton, D. ; Kennedy, J “Defining a Standard for Particle Swarm Optimization” Swarm Intelligence Symposium, 2007. SIS 2007. IEEE

[14] Ganesh Kumar, Salman Mohagheghi, Jean-Carlos Hernandez Yamille delValle, "Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems., " in IEEE, 2008, pp. 171-195.

[15] F. van den Bergh, A. P. E ngelbrecht “A New Locally Convergent Particle Swarm Optimiser” IEEE International Conference on Systems, Man and Cybernetics, 2002

[16] J. Kennedy. “Small Worlds and Mega-Minds: Effects of Neighborhood Topology on Particle Swarm Performance”. In Proceedings of the IEEE Congress onEvolutionary Computation, volume 3, pages 1931–1938, 1999.

[17] Satyobroto Talukder “ Mathematical Modelling and Applications of Particle Swarm Optimization” Master’s Thesis Mathematical Modelling and Simulation Thesis no: 2010: 8

[18] Morten Løvbjerg and Thiemo Krink, "Extending Particle Swarm Optimisers with Self-Organized Criticality, " IEEE Int. Congr. Evolutionary Computation, vol. 2, pp. 1588-1593, May 2002

[19] X. Xie, W. Zhang, and Z. Yang. A Dissipative Particle Swarm Optimization. In Proceedings of the IEEE Congress on Evolutionary Computation, volume 2, pages 1456–1461, 2002.

[20] X. Xie, W. Zhang, and Z. Yang. Adaptive Particle Swarm Optimization on Individual Level. In Proceedings of the Sixth International Conference on SignalProcessing, volume 2, pages 1215–1218, 2002.

[21] Massimiliano Kaucic "A multi-start opposition-based particle swarm optimization algorithm with adaptive velocity for bound constrained global optimization" J Glob Optim (2013) 55: 165–188

[22] Xiao-Feng Xie, Wen-Jun Zhang, Zhi-Lian Yang "Adaptive Particle Swarm Optimization on Individual Level" International Conference on Signal Processing (ICSP), Beijing, China, 2002: 1215-1218

[23] Ziyang Zhen, Daobo Wang, Meng Li "Improved Particle Swarm Optimizer Based on Adaptive Random Learning Approach" IEEE Congress on Evolutionary computation 2009

[24] Mengqi Hu, Teresa Wu, and Jeffery D. Weir "An Adaptive Particle Swarm Optimization With Multiple Adaptive Methods" IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 17, NO. 5, OCTOBER 2013

[25] Xueming Yang, Jinsha Yuan, Jiangye Yuan, Huina Mao "A modified particle swarm optimizer with dynamic adaptation" Applied Mathematics and Computation 189 (2007) 1205–1213

[26] B. Al-Kazemi and C. K. Mohan. Multi-Phase Generalization of the Particle Swarm Optimization Algorithm. In Proceedings of the IEEE Congress on EvolutionaryComputation, pages 489–494, 2002.

[27] B. Al-Kazemi and C. K. Mohan. Training Feedforward Neural Networks using Multi-Phase Particle Swarm Optimization. In Proceedings of the NinethInternational Conference on Neural Information Processing, volume 5, pages 2615–2619, 2002.

[28] B. Al-Kazemi and C. K. Mohan. Multi-Phase Discrete Particle Swarm Optimization. In Proceedings of the International Workshop on Frontiers in EvolutionaryAlgorithms, pages 622–625, 2002.

[29] L. M. Ruan, M. Shi, W. An and H. P. Tan H. Qi, "Application of multi-phase particle swarm optimization technique to inverse radiation problem, " Journal of Quantitative Spectroscopy & Radiative Transfer 109 (3), pp. 476-493, 2008.

[30] Minghua Li, Quan Liu, Wangshu Yao, Ming Chen " A Particle Swarm Optimization Based on improved Multi Swarm and Analysis" The 3rd Intetnational Conference on Innovative Computing Information and Control IEEE 2008

[31] Zhiqiang Geng and Qunxiong Zhu "A Multi-swarm PSO and its Application in Operational Optimization of Ethylene Cracking Furnace" Proceedings of

the 7th World Congress on Intelligent Control and Automation June 25 - 27, 2008

[32] Zhao Xinchao, "A perturbed particle swarm algorithm for numerical optimization, " Applied Soft Computing 10, pp. 119–124, 2010.

[33] Abdelaziz Laifa, Mohamed Boudour "Optimal Placement and Parameter Settings of Unified Power Flow Controller Device using a Perturbed Particle Swarm Optimization" IEEE International Energy Conference 2010

[34] Mehmet Sevkli, Aise Zulal Sevkli "A Stochastically Perturbed Particle Swarm Optimization for Identical Parallel Machine Scheduling Problems" Bio-Inspired Computational Algorithms and Their Applications. Intech open

[35] Lei Zhang, Ke Xu, Ruiqing Fu, Yongsheng Ou, and Xinyu Wu "A Stochastic Perturbing Particle Swarm Optimization Model" Second WRI Global Congress on Intelligent Systems 2010.

[36] Fulong Chen, Angdi Chu, Xuefei Yang, Yao Lei, Jizheng Chu "Identification of the Parameters of the Beeler Reuter Ionic Equation With a Partially Perturbed Particle Swarm Optimization" IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, VOL. 59, NO. 12, DECEMBER 2012

[37] Timothy Marler, Jasbir S. Arora "Survey of multi-objective optimization methods for engineering" Structural and Multi-disciplinary Optimization 26, 369–395 (2004)

[38] J. Moore and R. Chapman, Application of particle swarm to multiobjective optimization, Technical report, Department of Computer Science and Software Engineering, Auburn University, 1999.

[39] Wickramasinghe, W., Li, X. Integrating user preferences with particle swarms for multi-objective optimization. In GECCO'08, Atlanta, USA, pp. 745–752 (2008)

[40] R. Timothy Marler, Jasbir S. Arora "The weighted sum method for multi objective optimization: new insights" Structural and Multidisciplinary Optimization June 2010, Volume 41, Issue 6, pp 853-862

[41] Konstantinos E. Parsopoulos, Michael N. Vrahatis "Multi-Objective Particles Swarm Optimization Approaches" IGI Global 2008

[42] Carlos A. Coello Coello, Gregorio Toscano Pulido, Maximino Salazar Lechuga "Handling Multiple Objectives With Particle Swarm Optimization" IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 8, NO. 3, JUNE 2004

[43] L. B. Zhang', C. G Zhou', X. H. Liu', Z. Q. Ma'. ', M. Ma' and Y. C. Liang "Solving Multi Objective Optimization Problems Using Particle Swarm Optimization" Congress on Evolutionary Computation, Volume: 4, IEEE 2003.

[44] Masashi Yamamoto, Takeshi Uchitane, Toshiharu Hatanaka "An Experimental Study for Multi–objective Optimization by Particle Swarm with Graph Based Archive" SICE Annual Conference 2012

[45] Konstantinos E. Parsopoulos and Michael N. Vrahatis, "Multi-Objective Particles Swarm Optimization Approaches", IGI Global, 2008.

[46] S. Ganguly · N. C. Sahoo · D. DasA "novel multi-objective PSO for electrical distribution system planning incorporating distributed generation" Energy System (2010) 1: 291–337

[47] Leticia Cagnina, Susana Esquivel Carlos A. Coello Coello "A Particle Swarm Optimizer for Multi-Objective Optimization"JCS&T Vol. 5 No. 4 December 2005

[48] S. LALWANI_, S. SINGHAL, R. KUMAR AND N. GUPTA "A COMPREHENSIVE SURVEY: APPLICATIONS OF MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION (MOPSO) ALGORITHM" Transactions on Combinatorics Vol. 2 No. 1 (2013), pp. 39-101

[49] Nor Azlina Ab. Aziz, Mohamad Yusoff Alias, Ammar W. Mohemmed, Kamarulzaman Ab. Aziz, "Particle Swarm Optimization for Constrained and Multiobjective Problems: A Brief Review" International Conference on Management and Artificial Intelligence IPEDR vol. 6 (2011)

[50] C. S. Yang, L. Y. Chuang, and J. C. Li, "Chaotic maps in binary particle swarm optimization for feature selection, " in Proc. IEEE Conf. SMCIA, 2008, pp. 107–112.

[51] Bing Xue, Mengjie Zhang, Will N. Browne "Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach" IEEE TRANSACTIONS ON CYBERNETICS, VOL. 43, NO. 6, DECEMBER 2013

[52] X. Hu and R. Eberhart. Multiobjective optimization using dynamic neigborhood particle swarm optimization. In Congress on Evolutionary Computation, pages 1677{1681, Piscataway, NJ., 2002. IEEE Service Center.

[53] Xiaohui Hu, Russell Eberhart "Multiobjective Optimization Using Dynamic Neighborhood Particle Swarm Optimization" Congress on Evolutionary Computation, IEEE, 2002

[54] Pradipta Ghosh1, Hamim Zafar1, Swagatam Das1 and Ajith Abraham "Hierarchical Dynamic Neighborhood Based Particle Swarm Optimization for Global Optimization" Congress on Evolutionary Computation, IEEE, 2011

[55] S. Burak Akat, Veysel Gazi "Particle Swarm Optimization with Dynamic Neighborhood Topology: Three Neighborhood Strategies and Preliminary Results" IEEE Swarm Intelligence Symposium 2008.

[56] Alejandro Cervantes, Inés María Galván, and Pedro Isasi "AMPSO: A New Particle Swarm Method for Nearest Neighborhood Classification" IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 39, NO. 5, OCTOBER 2009

[57] Ehsan Safavieh, Amin Gheib, Mohammadreza Abolghasemi, and Ali Mohades "Particle Swarm Optimization with Voronoi Neighborhood" Proceedings of the 14th International CSI Computer Conference 2009.

[58] Aimin Miao, Xinling Shi, Junhua Zhang, Wei Jiang, Jinlin Zhang, Xiaolin Gui "Dynamic Particle Swarm Optimization Based on Neighborhood Rough Set Model" 2nd International Asia Conference on Informatics in Control, Automation and Robotics 2010.

[59] Min Han Jianchao Fan "Particle Swarm Optimization using Dynamic Neighborhood Topology for Large Scale Optimization" Proceedings of the 8th World Congress on Intelligent Control and Automation IEEE 2010

[60] PENG Hu, DENG Chang-shou "Dynamic Neighborhood Hybrid Particle Swarm Optimization for Constrained Opt imization" International Conference on Computational and Information Sciences, IEEE 2010

[61] Lily D Li, Xinghuo Yu, Xiaodong Lid William Guo "A Dynamic Neighbourhood Particle Swarm Optimisation Algorithm for Constrained Optimisation" 37th Annual Conference on IEEE Industrial Electronics Society, IEEE, 2011

[62] M. R. Pontes, C. J. A. Bastos-Filho R. A. Santana, "A Multiple Objective Particle Swarm Optimization Approach using Crowding Distance and Roulette Wheel, " IEEE Ninth International Conference on Intelligent Systems Design and Applications, pp. 237-242, 2009.

[63] Li Feng, Ziyan Liu, Chao Ma, Lin Huang, Li Zhao, "Multi-Objective Vector Evaluated PSO with Time Variant Coefficients for Outlier Identification in Power Systems" 43rd International Universities_Power_Engineering Conference, IEEE, 2008.

[64] K. E Parsopoulos, M. N. Vrahatls. "Recent approaches in global optimization problem through Particle Swarm Optimization" Natural computing 1: 235-305, 2002

[65] Dennis Gies and Yahya Rahmat-Samii "Vector Evaluated Particle Swarm Optimization (VEPSO): Optimization of a Radiometer Array Antenna" Antennas and Propagation Society International Symposium, 2004. IEEE

[66] Zuwairie Ibrahim, Noor Khafifah Khalid, Kian Sheng Lim, Salinda Buyamin, Jameel Abdulla Ahmed Mukred "A Binary Vector Evaluated Particle Swarm Optimization Based Method for DNA Sequence Design Problem" IEEE Student Conference on Research and Development 2011

[67] Lim Kian Sheng, Zuwairie Ibrahim, Salinda Buyamin, Anita Ahmad, Ismail Ibrahim, Mohd Falfazli "Convergence and Diversity Evaluation for Vector Evaluated Particle Swarm Optimization" Proceedings of International Conference on Modelling, Identification & Control (ICMIC) 2013.

[68] Ihsan Mohd Yassin, Mohd Nasir Taib, Hasliza Abu Hassan, Azlee Zabidi and Nooritawati Md. Tahir "Heat Exchanger Modeling using NARX Model with Binary PSO-based Structure Selection Method" International Conference on Computer Applications and Industrial Electronics IEEE 2010.

[69] A. Angelah Queen, D. Jeraldin Auxillia "Simplijied Discrete Binary PSO tuned Multivariable PID Controller for Binary Distillation Column Plant" International Conference on Circuits, Power and Computing Technologies IEEE 2013.

[70] Fatih Camci "Analysis of Velocity Calculation Methods in Binary PSO on Maintenance Scheduling" First International Conference on the Applications of Digital Information and Web Technologies, IEEE, 2008.

[71] LanLan Zhen, Ling Wang, Ziyuan Huang "Probability-based Binary Particle Swarm Optimization Algorithm and Its Application to WFGD Control"

International Conference on Computer Science and Software Engineering IEEE 2008.

[72] LanLan Zhen, Ling Wang, Xiuting Wang, Ziyuan Huang "A Novel PSO-inspired Probability-based Binary Optimization Algorithm" International Symposium on Information Science and Engieering IEEE 2008

[73] Arezoo Modiri, Kamran Kiasaleh, "Modification of Real-Number and Binary PSO Algorithms for Accelerated Convergence" IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION, VOL. 59, NO. 1, JANUARY 2011

[74] Shirin Khezri1, Karim Faez 2, Amjad Osmani "Modified Discrete Binary PSO based Sensor Placement in WSN Networks" International Conference on Computational Intelligence and Communication Networks IEEE 2010

[75] Vahid Hosseinnezhad, Ebrahim Babaei "Economic load dispatch using h-PSO" Electrical Power and Energy Systems 49 (2013) 160–169.

[76] Wei-min ZHONG1, 2, Shao-jun LI1, 2, Feng QIAN "θ-PSO: a new strategy of particle swarm optimization" Zhong et al. / J Zhejiang Univ Sci A 2008 9(6): 786-790

[77] Amin Safari "Θ-PSO Algorithm for UPFC Based Output Feedback Damping Controller" International Journal of Control and Automation Vol. 6, No. 1, 2013

[78] Weimin Zhong Jianliang Xing Feng Qian "An improved theta-PSO algorithm with crossover and mutation" Proceedings of the 7th World Congress on Intelligent Control and Automation, IEEE 2008.

[79] K. Pouya, H. Lesani "An Angle-Based PSO Approach For Reactive Power Management Problem" Power Systems Conference and Exposition, IEEE, 2009

[80] Siavash Beheshtaein "FAΘPSO Based Fuzzy Controller to Enhance LVRT Capability of DFIG with Dynamic References" 23rd International Symposium on Industrial Electronics (ISIE), IEEE, 2014.

[81] F. van den Bergh and A. P. Engelbrecht. Cooperative Learning in Neural Networks using Particle Swarm Optimizers. South African Computer Journal, (26): 84–90, November 2000. 21, 134, 143

[82] F van den Bergh, A. P. Engelbrecht "Training Product Unit Networks using Cooperative Particle Swarm Optimisers" International Joint Conference on Neural Networks, IEEE, 2001.

[83] I. Ziari, A. Jalilian "A PSO-Based Approach to Adjust CPSO Parameters" 14th International Conference on Harmonics and Quality of Power, IEEE, 2010

[84] LIU Lei, JING Tao, JIA Lu "Research of Load-Balance Technique for Cooperative PSO" International Conference on Electrical and Control Engineering IEEE, 2010.

[85] Nikolas J. Unger, Beatrice M. Ombuki-Berman, Andries P. Engelbrecht "Cooperative Particle Swarm Optimization in Dynamic Environments" Symposium on Swarm Intelligence IEEE2013

[86] Frans van den Bergh and Andries P. Engelbrecht "A Cooperative Approach to Particle Swarm Optimization", IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 8, NO. 3, JUNE 2004

[87] Fuqiang Xie, Ye Yang, Songtao Chang, and Yongji Wang "Improving Strategies on PSO for Suborbit Launch Vehicle Trajectory Optimization" Fourth International Workshop on Advanced Computational Intelligence IEEE, 2011.

[88] J. Sun, B. Feng, and W. Xu, "Particle swarm optimization with particles having quantum behavior, " in Proc. Congr. Evol. Comput., 2004, pp. 325–331.

[89] Jin Yanxia, Zhan Hanchang "An improved quantum particle swarm optimization algorithm" 2nd International Conference onInformation Science and Engineering, IEEE, 2010.

[90] Qun Zhang, Xiujuan Lei1, Xu Huang, Aidong Zhang "An Improved Projection Pursuit Clustering Model and its Application Based on Quantum-behaved PSO" Sixth International Conference on Natural Computation IEEE, 2010.

[91] Wei Zhao, Ye San and Huishu Shi "Fuzzy Quantum-Behaved Particle Swarm Optimization Algorithm" International Symposium on Computational Intelligence and Design, IEEE, 2010.