Particle Swarm Optimization: Technique, System and Challenges

Dian Palupi Rini
Faculty of Computer Science,
Sriwijaya University, Indonesia
Faculty of Computer Science &
Information System,
Universiti Teknologi Malaysia

Siti Mariyam Shamsuddin
Faculty of Computer Science &
Information System,
Universiti Teknologi Malaysia

Siti Sophiyati Yuhaniz
Faculty of Computer Science &
Information System,
Universiti Teknologi Malaysia

ABSTRACT
Particle Swarm Optimization (PSO) is a biologically inspired computational search and optimization method developed in 1995 by Eberhart and Kennedy based on the social behaviors of birds flocking or fish schooling. A number of basic variations have been developed due to improve speed of convergence and quality of solution found by the PSO. On the other hand, basic PSO is more appropriate to process static, simple optimization problem. Modification PSO is developed for solving the basic PSO problem. The observation and review 46 related studies in the period between 2002 and 2010 focusing on function of PSO, advantages and disadvantages of PSO, the basic variant of PSO, Modification of PSO and applications that have implemented using PSO. The application can show which one the modified or variant PSO that haven't been made and which one the modified or variant PSO that will be developed.

Keywords
Particle Swarm Optimization (PSO), Variant PSO, Modification PSO, Basic PSO problem, Bird Flocking, Evolutionary Optimization, biologically inspired computational search.

1. INTRODUCTION
Theory of particle swarm optimization (PSO) has been growing rapidly. PSO has been used by many applications of several problems. The algorithm of PSO emulates from behavior of animals societies that don't have any leader in their group or swarm, such as bird flocking and fish schooling. Typically, a flock of animals that have no leaders will find food by random, follow one of the members of the group that has the closest position with a food source (potential solution). The flocks achieve their best condition simultaneously through communication among members who already have a better situation. Animal which has a better condition will inform it to its flocks and the others will move simultaneously to that place. This would happen repeatedly until the best conditions or a food source discovered. The process of PSO algorithm in finding optimal values follows the work of this animal society. Particle swarm optimization consists of a swarm of particles, where particle represent a potential solution.

Recently, there are several modifications from original PSO. It modifies to accelerate the achieving of the best conditions. The development will provide new advantages and also the diversity of problems to be resolved. Study on the development of PSO is necessary to do to know how far its development, its advantages and disadvantages and how much use this method to settle a problem. Tutorial and theoretical of PSO has made about what is PSO [1], [2], those describe about what PSO is, simple data tested, and comparison with others evolutionary computations.

This paper will describe what for the modifications, advantages and disadvantages each modification of PSO and make a conclusion from those. In section 2 describes about basic PSO, basic variation of PSO, and modification of PSO, and section 3 observation, and the last section 4 describe about summary and future work.

2. VARIANT OF PSO
Exploration is the ability of a search algorithm to explore different region of the search space in order to locate a good optimum. Exploitation, on the other hand, is the ability to concentrate the search around a promising area in order to refine a candidate solution[3].With their exploration and exploitation, the particle of the swarm fly through hyperspace and have two essential reasoning capabilities: their memory of their own best position - *local best (lb)* and knowledge of the global or their neighborhood's best - *global best (gb)*.

Position of the particle is influenced by velocity. Let $x_i(t)$ denote the position of particle i in the search space at time step t; unless otherwise stated, t denotes discrete time steps. The position of the particle is changed by adding a velocity, $v_i(t)$ to the current position [1]:

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

where [2] :

$$v_i(t) = v_i(t-1) + c_1 r_1\big(localbest(t) - x_i(t-1)\big) + c_2 r_2\big(globalbest(t) - x_i(t-1)\big) \tag{2}$$

with $x_i(0) \sim U\,(x_{min}, x_{max})$, acceleration coefficient c_1 and c_2, and random vector r_1 and r_2. Simple example of PSO, there is a function [3]:

$$\text{Min}\, f(x)$$

$$\text{where } x(B) \leq x < x(A)$$

Denote $x(B)$ as a lower limit and $x(A)$ as an upper limit. So, PSO procedure can be described by the following steps: *First*, Assume that the size of the group of particle is N. It is necessary that the size N is not too large, but also not too small, so that there are many possible positions toward the best solution or optimal. *Second*, generate initial population x with range $x(B)$ and $x(A)$ by random order to get the $x_1, x_2, \ldots, x_n$. It is necessary if the overall value of the particle is uniformly in the search area.

After that, the particle j and the velocity at iteration i are denoted as $x_j(i)$ and $v_j(i)$. thus, these initial particles will be $x_1(0), x_2(0), \ldots, x_n(0)$. Vector $x_j(0)$, $(j = 1,2, \ldots, n)$ is called a particle or vector coordinates of the particle. (Such as: chromosomes in genetic algorithms). Evaluation of the objective function value for each particle and expressed by $f[x_1(0)], f[x_2(0)], \ldots, f[x_n(0)]$

Then calculate the speed of all particles. All particles move towards the optimal point with a velocity. Initially all of the particle velocity is assumed to be zero. Set iteration $i = 1$.

At the i^{th} iteration, find the two important parameters for each particle j that is:

a. The best value of $x_j(i)$ (the coordinates of particle j at iteration i) and declare as $P_{best}(j)$, with the lowest value of objective function (minimization case), $f\,[x_j\,(i)]$, which found a particle j at all previous iteration. The best value for all particles $x_j(i)$ which found up to the i^{th} iteration, G_{best} with the value function the smallest goal / minimum among all particles for all the previous iterations, $f[x_j(i)]$.
b. Calculate the velocity of particle j at iteration i using the following formula using formula (2):
Where c_1 and c_2, respectively, are learning rates for individual ability (cognitive) and social influence (group), and r_1 and r_2 uniformly random numbers are distributed in the interval 0 and 1. So the parameters c_1 and c_2 represent weight of memory (position) of a particle towards memory (position) of the groups (swarm). The value of c_1 and c_2 is usually 2, so multiply $c_1 r_1$ and $c_2 r_2$ ensure that the particles will approach the target about half of the difference
c. Calculate the position or coordinates of particle j at the i^{th} iteration by :
$x_i(t+1) = x_i(t) + v_i(t+1)$

Evaluation of the objective function value for each particle and expressed as: $f[x_1(i)], f[x_2(i)], \ldots, f[x_n(i)]$

The last step, check whether the current solution is convergent. If the positions of all particles leading to an equal value, then this is called convergence. If not convergent then step 4 is repeated by updating iterations $i = i + 1$, by calculating new values from $P_{best}(j)$ and G_{best}. This iteration process continues until all particles convergence the same solution. Usually be determined by the termination criteria (Stopping criterion), for example the amount of the excess solution with a solution now previously been very small.

If the current solution is convergent, then the iteration will stop. We do not know whether the final value is the best value. Below are the stopping criteria conditions for the iteration: *First*, terminate when a maximum number of iterations, or FEs, has been exceeded. *Second*, Terminate when an acceptable solution has been found, *Third*, Terminate when no improvement is observed over a number of iteration. *Fourth*, terminate when the normalized swarm radius is close to zero. *Fifth*, terminate when the objective function slope is approximately zero. Although the particle has stopped, we do not know whether the particle will pitch on local optima, local minima, global optima or global optima.

In the original particle swarm optimization, there has also a lack of solution, because it is very easy to move to *local optima.* In certain circumstances, where a new position of the particle equal to global best and local best then the particle will not change its position. If that particle is the global best of the entire swarm then all the other particles will tend to move in the direction of this particle. The end of result is the swarm converging prematurely to a local optimum. If the new position of the particle pretty far from global best and local best then the velocity will changing quickly turned into a great value. This will directly affect the particle's position in the next step. For now the particle will have an updated position of great value, as a result, the particle may be out of bounds the search area.

In analysis, PSO has advantages and disadvantages [4]. Advantages of the basic particle swarm optimization algorithm: PSO is based on the intelligence. It can be applied into both scientific research and engineering use. Then PSO have no overlapping and mutation calculation. The search can be carried out by the speed of the particle. During the development of several generations, only the most optimist particle can transmit information onto the other particles, and the speed of the researching is very fast. After that the calculation in PSO is very simple. Compared with the other developing calculations, it occupies the bigger optimization ability and it can be completed easily. The last one is PSO adopts the real number code, and it is decided directly by the solution. The number of the dimension is equal to the constant of the solution.

On the other hands, disadvantages of the basic particle swarm optimization algorithm are the method easily suffers from the partial optimism, which causes the less exact at the regulation of its speed and the direction. Then the method cannot work out the problems of scattering and optimization and the method cannot work out the problems of non-coordinate system, such as the solution to the energy field and the moving rules of the particles in the energy field.

2.1 Basic Variants of PSO

The lacks of PSO have been reduced with a variation of PSO. Many variations have been developed to improve speed of convergence and quality of solution found by the PSO. The variation is influenced by a number of control parameters, namely the dimension of the problem, the number of particles (swarm size), acceleration coefficients (The acceleration coefficient, c_1 and c_2, together with random vector r1 and r2, control the stochastic influence), inertia weight, neighborhood size, number of iteration, and the random values which scale the contribution of the cognitive and social component. Below are the basic variations of particle swarm optimization:

a. Velocity clamping

Velocity clamping will control the global exploration of the particle. If the velocity v of a particle i exceeds the maximum allowed speed limit, it will set a maximum value of velocity (v_{max} (j)). So that v_{max}, j indicates the maximum allowable speed for a particle in the j^{th} dimension. Speed (velocity) of the particle is adjusted using the equation [2]:

$$v_{ij} = \begin{cases} v'_{ij}(t+1), if\ v_{ij}(t+1) < v_{max}(J) \\ v_{max}(j)\ \ otherwise \end{cases} \tag{3}$$

High value of v_{max} (j) will cause global exploration, whereas lower values result in local exploration. v_{max} (j) will control the movement of the particle and aspect of exploration and exploitation. Velocity clamping did not influence the position of the particle. This only reduces the size of the step velocity. Changes in the search direction not only can make a particle to perform a better exploration but also has negative effects and the optimum value cannot be found.

The following equation [2] is used to initialize the max and min velocity to the solution:

$$v_{max,j} = \delta\ (x_{max,j} - x_{min,j}) \tag{4}$$
$$v_{min,j} = \delta\ (x_{min,j} - x_{max,j}) \tag{5}$$

Where as $x_{max,j}$ and $x_{min,j}$ are the minimum and maximum positions of the particle in the j thdimension. δ is a constant factor and is taken from 0 until 1. The problem is if all the velocity becomes equal to v_{max} the particle will continue to conduct searches within a hypercube and will probably remain in the optima but will not converge in the local area.

There are some researchers that have develop velocity clamping method, such as : [5], [6]

b. Inertia weight

It is a mechanism to control an exploration and exploitation abilities of the swarm, and as mechanism to eliminate the need of velocity clamping. 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. For the $gbest$ PSO, the velocity equation [5] changes from equation:

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_{1j}(t)\left(y_{ij}(t) - x_{ij}(t)\right) + c_2 r_{2j}(t)\left(\hat{y}_j(t) - x_{ij}(t)\right) \tag{6}$$

A similar change is made from the-$lbest$ PSO. Inertia weight presenting how much the amount of memory from the previous flight direction will affect the new velocity. If $w > 1$, then the velocity will decrease with time, the particle will accelerate to maximum velocity and the swarm will be divergent. If $w < 1$, then the velocity of particle will decrease until it reaches zero. The larger value of w will facilitate an exploration, rather small values will promote the exploitation.There are some researchers that have develop inertia weight application, such as :[7], [8], [9], [10]

c. Constriction Coefficient

Velocity update equation that using constriction coefficient changes to:

$$v_{ij}(t+1) = x\left[v_{ij}(t) + \emptyset_1(y_{ij}(t) - x_{ij}(t) + \emptyset_2(\hat{y}_j(t) - x_{ij}(t)\right] \tag{7}$$

Where

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

With

$$\phi = \phi_1 + \phi_2$$
$$\phi_1 = c_1 r_1$$
$$\phi_2 = c_2 r_2$$

Equation above is used under the constraints that $\emptyset \geq 4\ and\ k \in [0,1]$. The constriction approach was developed as a natural, dynamic way to ensure convergence to a stable point, without the need for velocity clamping. Condition $\emptyset \geq 4$ and $k \in [0,1]$ of the swarm is guaranteed to convergence.

There are some researchers that have develop constriction coefficient , such as : [11], [12].

d. Synchronous Versus Asynchronous Updates

Synchronous Updates [13] are done separately from the particle (personal best and neighborhood bests) position updates, only given one feedback per iteration update, slower feedback and better for $gbest$. While asynchronous is better for $lbest$, updates calculate the new best positions after each particle position update and have the advantage that immediate feedback is given about the best region of search space. There are some researchers that have develop this method, such as : [14], [15], [16], [17], [18], [19], [20].

2.2. Modification of PSO

The modification in PSO consists of three categories: extension of field searching space, adjustment the parameters, and hybrid with another technique. The modifications of PSO can enhance its performance.

a. Single Solution PSO

A large number of PSO variations can be found to locate single solutions. These PSO implementations were specially developed to obtain single solutions to continuous-valued, unconstrained, static, single-objective, optimization problem, most of these algorithm can also be applied to other problem types.

b. Niching with PSO

In the EC field, algorithms that locate multiple solutions are refers to as niching algorithm. The process of finding multiple solution or niche is generally referred to as speciation. Niching algorithms model yet another natural process, where large numbers of individuals compete for the use of limited resources on physical environment.

Nieces are partitions of an environment while species are partitions of computational optimization, a niece represents one solutions to the problem, while a species refers to the group of individuals (particle in the context of PSO) that convergence on a single niece.

c. Constraint Optimization using PSO

Constraint reduces the feasible space where in solution to the problem can be found. Optimization algorithms need to ensure that a feasible solution is found. That is the optimization algorithm should find a solution that both optimizes the objective function satisfies all constraints. If it is not possible to satisfy all constrains, the algorithm has to balance the trades off between optimal objective function value and number of constrain violated.

d. Multi-objective optimization with PSO

Many real world optimization problems require the simultaneous optimization of a number of objectives (multi-objectives). Using the notation, the multi-objectives optimization problem is defined as:

$$\begin{aligned} &minimize \;\; f(x), \;\; x = (x_1, x_2, \dots, x_n) \\ &subject\ to \;\; g_m \leq 0, \quad m = 1, \dots, n_g \\ &\qquad\qquad h_m = 0, \quad m = n_g + 1, \dots, n_g + n_h \end{aligned} \tag{8}$$

The main objective of MOO algorithms is to find a set of solution which optimally balance the trade-offs among the objective of a MOP. It is different with the basic PSO that return only one solution.

e. Dynamic Environment With PSO

In dynamic Environments, PSO should be fast to allow quick re-optimization. It is desirable to find a good solution before the next environment change. In original PSO, it is impossible to convergence to an equilibrium state in its first goal to locate the optimum.

There are several solutions for dynamic environment. Such as: a. Environment change detection, It is to allow timeous and efficient tracking of optimum, b. Response to environment changes, c. Changing the inertia weight update, d. Reinitialize Particle Solution, e. Limit Memory, f. Local Search, g. Split adaptive PSO, h. Fine-Grained, i. charged Swarm, The changed PSO charges the velocity equation by adding a particle acceleration, a_i, to the standard equation, That is:

$$\begin{aligned} v_{ij}(t+1) = \omega v_{ij}(t) + c_1 r_1(t)\left[y_{ij}(t) - x_{ij}(t)\right] \\ + c_2 r_2(t)\left[\hat{y}_j(t) - x_{ij}(t)\right] + a_{ij}(t) \end{aligned}$$

Where: $a_i(t) = \sum_{l=1, i\neq l}^{n_s} a_{il}$ (9)

f. Discrete PSO

PSO was originally developed for continuous-valued spaces. Many problems are, however, defined for discrete value. Fortunately, the PSO is easily adaptable to discrete-value spaces.

i. Binary PSO

For the binary PSO [1], particle represents position in binary space. Formally, element of a particle's position (x_i) can form as: $x_i \in \mathbb{B}^{n_x}$ on $x_{ij} \in \{0,1\}$. A natural normalization of velocities is obtained by using sigmoid function, that is:

$$v_{ij}(t) = sig\left(v_{ij}(t)\right) = \frac{1}{1 + e^{-v_{ij}(t)}} \tag{10}$$

The position update changes to:

$$x_{ij}(t+1) = \begin{cases} 1 & if\ r_{3j}(t) < sig\left(v_{ij}(t+1)\right) \\ 0 & other \end{cases} \tag{11}$$

Where $r_{3j}(t) \sim U(0,1)$. Many applications have used binary PSO to solve their problem.

ii. General Discrete

Clerk defines these operators for participles that represent a permutation of the valid discrete values with a strong ordering implied between dimensions. In general velocity and position equation change to:

$$v_i(t+1) = \left(\omega \otimes v_i(t)\right) \circ \left(c_1 r_1(t)\left(y_i(t)\right) \ominus x_i(t)\right) \circ \left(\left(c_1 r_1(t)\left(\hat{y}_i(t)\right) \ominus x_i(t)\right) \tag{12}$$

Position update: first, all velocities are normalized to the range [0,1] by dividing the velocities by the maximum range of the corresponding dimensions,

$$\begin{aligned} x_i(t+1) &= x_i(t) \oplus v_i(t+1) \\ &= \left(\omega \otimes v_i(t)\right) \end{aligned} \tag{13}$$

Then, each position determines if there is a swap with probability v_{ij}. Last, if a swap has to be executed, the affected position of the particle change to that of the global best (or local best) position.

3. OBSERVATION AND REVIEW

Particle swarm optimization (PSO) is a biologically inspired computational search and optimization method developed in 1995 by Eberhart and Kennedy based on the social behaviors of birds flocking or fish schooling. Recently, there are many variants of PSO, and it may always grow rapidly. Figure 1 describes the variants of particle swarm.

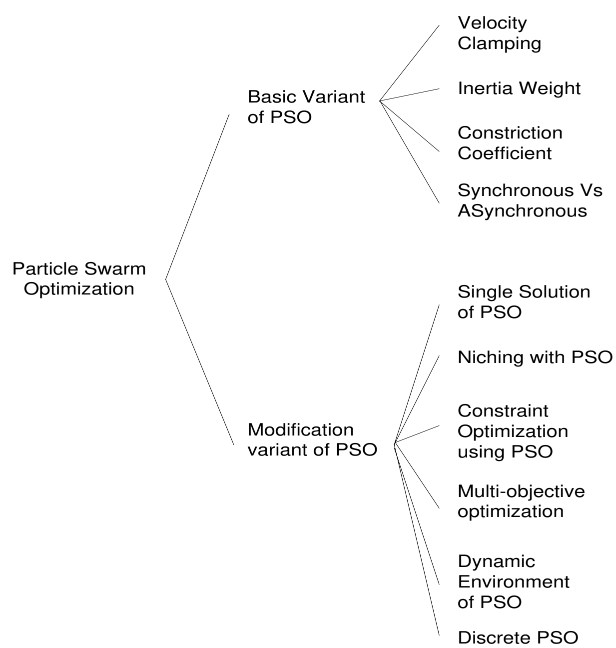

Fig 1: Variant of Particle Swarm Optimization

We have considered that velocity clamping, inertia weight, constriction coefficient, synchronous and asynchronous updates are the basic variations of PSO that have been developed to improve speed of convergence and quality of solution found by the PSO. Figure 2 presents distribution of articles in terms of basic variant of PSO. Regarding on this inertia weight has the largest number of literatures between 2006 and 2010. Due to the progress of variant PSO is rather new, so there is only a few articles that has made.

Every basic variant of PSO has utility that will cover shortfall of PSO. In addition they also have advantages and disadvantages as shown in the table below:

Table1. The Basic Variant of PSO

Basic Variant	Function	Advantages	Disadvantages
Velocity Clamping	Control the global exploration of the particle Reduces the size of the step velocity, so that the particles remain in the search area, but it cannot change the search direction of the particle	VC reduces the size of the step velocity so it will control the movement of the particle	If all the velocity becomes equal to v_{max} the particle will continue to conduct searches within a hypercube and will probably remain in the optima but will not converge in the local area.
Inertia Weight	Controls the momentum of the particle by weighing the contribution of the previous velocity,	A larger inertia weight in the end of search will foster the convergence ability.	Achieve optimality convergence strongly influenced by the inertia weight
Constriction Coefficient	To ensure the stable convergence of the PSO algorithm [21]	Similar with inertia weight	when the algorithm converges, the fixed values of the parameters might cause the unnecessary fluctuation of particles
Synchronous and Asynchronous Updates	Optimization in parallel processing	Improved convergence rate	Higher throughput: More sophisticated finite element formulations Higher accuracy (mesh densities)

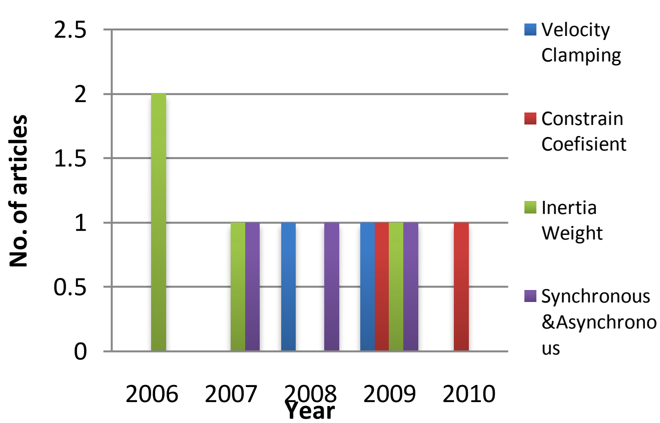

Fig 2.Distribution of articles for Basic variant of PSO

In this paper we have know that originally, particle swarm optimization is used to solve statics problem. For solving another form of problem, many researchers have developed variant PSO, such as: Single Solution, Niching with PSO, Constraint Optimization using PSO, Multi-objective optimization, Dynamic Environment and Discrete PSO. Every variant of PSO have different form and function. Each of them also has variety methods to solve their problem. Table 2 describes every characteristics of basic variant of PSO. There are many researchers that have develop many application using modification PSO. Figure 3 presents distribution of articles in terms of modification of Particle Swarm Optimization. The number of papers using single solution PSO yields a peak in 2007 and decreases gradually after that. Niching with PSO is only used by some of researchers. From the figure below, dynamic environment of PSO and multi-objective optimization are the bigger numbers of literatures between 2006 and 2010. But a number of article of dynamic environment decrease in 2010. On the others hands, the used of multi-objective optimization increase from time to time. This method has a challenge to increase caused of it can optimized multi-purposes of problems.

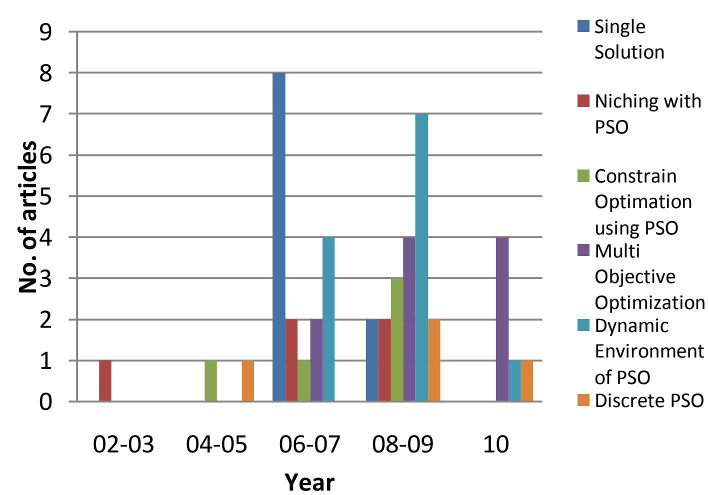

Fig 3.Distribution of articles in the term of modification of PSO

Table 2.Characteristic Modifications of PSO

Variant PSO	Utilities	Methods
Single Solution of PSO	Obtain single solutions to continuous-valued, unconstrained, static, single-objective, optimization problem	Social network structure, hybrid algorithm, sub-swarm-based, revealing methods, memetic PSO multi-start PSO
Niching with PSO	Niching (speciation) techniques have the ability to locate multiple solutions in multimodal domains	Quasi-sequential niching, Parallel niching algorithm, Objective function stretching, Sequential niching
Constra-int Optimization using PSO	Find a solution that both optimizes the objective function satisfies all constraints. If it is not possible to satisfy all constrains, the algorithm has to balance the trades off between optimal objective function value and number of constrain violated	convert to unconstrained problem, Repair method, Boundary constrain, Pareto ranking, Preserving feasible
Multi-objective optimization (MOO)	Find a set of solution among the objective of a multi optimization problem.	Criterion-based methods, dominance-base;
Dynamic Environment of PSO	Have an ability to solve an optimization in the dynamic real-world problems although if it is in multi objective optimization	Environment change detection, Response to environment changes, Changing the inertia weight update, Reinitialize Particle Solution, Limit Memory, Local Search, Split adaptive PSO, Fine-Grained, Charged Swarm
Discrete PSO	Find an optimization problem that operate on binary search space	Binary PSO, General Discrete PSO

With the characteristic of modification of PSO, there are several application areas that can develop, such as scheduling, searching, forecasting, feature selection, classification, production rate and functions problem. Table 3 described the distribution of article for every function of PSO.

Table 3.Distribution of article for every function of PSO

	Optimized Scheduling	Optimized Local Search	Optimized Multi Search	Optimized Forecasting	Optimized Function Problem	Optimized Feature Selection	Optimized Classification	Optimized Production Rate
Single Solution of PSO	[22] [23]	[24], [25] [26], [27]	[28]	[29]	[30]		[31]	
Niching With PSO		[32]	[33], [34] [35]		[36]			
Constrain Optimzed using PSO		[37]	[38]		[39] [40]			[41]
Multi Objective Optima zation	[42]	[43] [10]	[44]	[45]	[46]		[47] [48] [49]	[50]
Dinamic Environment of PSO	[51]	[52] [53]			[54], [55] [51]	[56], [51] [57]		[58], [59] [60]
Discrete PSO	[61]				[62]	[63]	[64]	

Modification of particle swarm optimization problems have implemented in several areas, i.e. Searching, Optimization mathematical function, Classification problem, Feature selection, Scheduling, and etc. Although the method of modification PSO has developed in many variant, it is very conducive to the creation of a new method of variation PSO because there are others area that can be implemented by modification PSO.

4. SUMMARY

The process of PSO algorithm in finding optimal values follows the work of an animal society which has no leader. Particle swarm optimization consists of a swarm of particles, where particle represent a potential solution (better condition). Particle will move through a multidimensional search space to find the best position in that space (the best position may possible to the maximum or minimum values).

In this paper, we have made review of the different methods of PSO algorithm. Basic particle swarm optimization has advantages and disadvantages, to overcome the lack of PSO. There are several basic variant of PSO. The basic variants as mentioned above have supported controlling the velocity and the stable convergence. At the other hands, modified variant PSO help the PSO to process other conditions that cannot be solved by the basic PSO.

The observation and review is made to show the absolute function of PSO, advantages and disadvantages of PSO, the basic variant of PSO, Modification of PSO and applications that have implemented using PSO. The application can show which one the modified or variant PSO that haven't been made and which one the modified or variant PSO that will be developed.

5. REFERENCES

[1] A. P. Engelbrecht, *Fundamental of Computational Swarm Inteligent*, First ed. The atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England: John Wiley & Sons Ltd, 2005.

[2] B. Santosa, "Tutorial Particle Swarm Optimization," 2006.

[3] M. B. Ghalia, "Particle Swarm Optimization with an Improved Exploration-Exploitation Balance," *iEEE,* vol. 978-1-4244-2167-1/08/$25.00 ©2008 IEEE, 2008.

[4] Q. Bai, "Analysis of Particle Swarm Optimization Algorithm," *Computer and Information Science,* vol. volume 3 No 1, Pebruari 2010 2010.

[5] F. Shahzad*, et al.*, "Opposition-Based Particle Swarm Optimization with Velocity Clamping (OVCPSO), ," *Journal advances in computational Intelligent, AISC 61, pp, 339-2348,* 2009.

[6] M. Ben Ghalia, "Particle swarm optimization with an improved exploration-exploitation balance," in *Circuits and Systems, 2008. MWSCAS 2008. 51st Midwest Symposium on*, 2008, pp. 759-762.

[7] A. Chatterjee and P. Siarry, "Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization," *Computers & Operations Research,* vol. 33, pp. 859-871, 2006.

[8] L. Yufeng, "Dynamic Particle Swarm Optimization Algorithm for Resolution of Overlapping Chromatograms," in *Natural Computation, 2009. ICNC '09. Fifth International Conference on*, 2009, pp. 246-250.

[9] S. Xianjun*, et al.*, "A Dynamic Adaptive Particle Swarm Optimization for Knapsack Problem," in *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*, 2006, pp. 3183-3187.

[10] P. K. Tripathi*, et al.*, "Multi-Objective Particle Swarm Optimization with time variant inertia and acceleration coefficients," *Information Sciences,* vol. 177, pp. 5033-5049, 2007.

[11] K. T. Chaturvedi*, et al.*, "Particle swarm optimization with time varying acceleration coefficients for non-convex economic power dispatch," *International Journal of Electrical Power & Energy Systems,* vol. 31, pp. 249-257, 2009.

[12] P. Boonyaritdachochai*, et al.*, "Optimal congestion management in an electricity market using particle swarm optimization with time-varying acceleration coefficients," *Computers & Mathematics with Applications,* vol. In Press, Corrected Proof, 2010.

[13] A. Engelbrecht, "particle Swarm Optimization : Pitfalls and convergen aspect."

[14] V. Kalivarapu*, et al.*, "Synchronous parallelization of Particle Swarm Optimization with digital pheromones," *Advances in Engineering Software,* vol. 40, pp. 975-985, 2009.

[15] S. B. Akat and V. Gazi, "Decentralized asynchronous particle swarm optimization," in *Swarm Intelligence Symposium, 2008. SIS 2008. IEEE*, 2008, pp. 1-8.

[16] V. Gazi, "Asynchronous Particle Swarm Optimization," in *Signal Processing and Communications Applications, 2007. SIU 2007. IEEE 15th*, 2007, pp. 1-4.

[17] I. Scriven*, et al.*, "Asynchronous multiple objective particle swarm optimisation in unreliable distributed environments," in *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, 2008, pp. 2481-2486.

[18] W. Bo*, et al.*, "Distributed Rate Allocation and Performance Optimization for Video Communication Over Mesh Networks," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, 2007, pp. VI - 501-VI - 504.

[19] Q. Liguo*, et al.*, "Design and Implementation of Intelligent PID Controller Based on FPGA," in *Natural Computation, 2008. ICNC '08. Fourth International Conference on*, 2008, pp. 511-515.

[20] T. Desell*, et al.*, "Robust Asynchronous Optimization for Volunteer Computing Grids," in *e-Science, 2009. e-Science '09. Fifth IEEE International Conference on*, 2009, pp. 263-270.

[21] L. T. Bui*, et al.*, "A Modified Strategy for the Constriction Factor in Particle Swarm Optimization," in *Book Series Lecture Notes in Computer Science* vol. Volume 4828/2010, ed. Heidelberg: Springer Berlin, 2010, pp. 333-344.

[22] L. Bo*, et al.*, "An Effective PSO-Based Memetic Algorithm for Flow Shop Scheduling," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 37, pp. 18-27, 2007.

[23] L. Zhixiong and W. Shaomei, "Hybrid Particle Swarm Optimization for Permutation Flow Shop Scheduling," in *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*, 2006, pp. 3245-3249.

[24] L. Dasheng*, et al.*, "A Multiobjective Memetic Algorithm Based on Particle Swarm Optimization," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 37, pp. 42-50, 2007.

[25] Y. G. Petalas*, et al.*, "Enhanced Learning in Fuzzy Simulation Models Using Memetic Particle Swarm Optimization," in *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE*, 2007, pp. 16-22.

[26] O. Schutze*, et al.*, "A Memetic PSO Algorithm for Scalar Optimization Problems," in *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE*, 2007, pp. 128-134.

[27] L. Hong-qi and L. Li, "A Novel Hybrid Particle Swarm Optimization Algorithm Combined with Harmony Search for High Dimensional Optimization Problems," in *Intelligent Pervasive*

Computing, 2007. IPC. The 2007 International Conference on, 2007, pp. 94-97.

[28] M. R. AlRashidi and M. E. El-Hawary, "Hybrid Particle Swarm Optimization Approach for Solving the Discrete OPF Problem Considering the Valve Loading Effects," *Power Systems, IEEE Transactions on,* vol. 22, pp. 2030-2038, 2007.

[29] Z. Ruiyou and W. Dingwei, "Forecasting annual electricity demand using BP neural network based on three sub-swarms PSO," in *Control and Decision Conference, 2008. CCDC 2008. Chinese*, 2008, pp. 1409-1413.

[30] J. Zhang*, et al.*, "Multi-sub-swarm particle swarm optimization algorithm for multimodal function optimization," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, 2007, pp. 3215-3220.

[31] L. Benameur*, et al.*, "A New Hybrid Particle Swarm Optimization Algorithm for Handling Multiobjective Problem Using Fuzzy Clustering Technique," in *Computational Intelligence, Modelling and Simulation, 2009. CSSim '09. International Conference on*, 2009, pp. 48-53.

[32] S. Changyin*, et al.*, "Clustering with a Weighted Sum Validity Function Using a Niching PSO Algorithm," in *Networking, Sensing and Control, 2007 IEEE International Conference on*, 2007, pp. 368-373.

[33] A. P. Engelbrecht and L. N. H. van Loggerenberg, "Enhancing the NichePSO," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, 2007, pp. 2297-2302.

[34] W. Junnian*, et al.*, "Hill Valley Function Based Niching Particle Swarm Optimization for Multimodal Functions," in *Artificial Intelligence and Computational Intelligence, 2009. AICI '09. International Conference on*, 2009, pp. 139-144.

[35] A. Nickabadi*, et al.*, "DNPSO: A Dynamic Niching Particle Swarm Optimizer for multi-modal optimization," in *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, 2008, pp. 26-32.

[36] R. Brits*, et al.*, "Solving systems of unconstrained equations using particle swarm optimization," in *Systems, Man and Cybernetics, 2002 IEEE International Conference on*, 2002, p. 6 pp. vol.3.

[37] M. Qianzhi*, et al.*, "Mobile Robot Path Planning with Complex Constraints Based on the Second-Order Oscillating Particle Swarm Optimization Algorithm," in *Computer Science and Information Engineering, 2009 WRI World Congress on*, 2009, pp. 244-248.

[38] M. R. AlRashidi and M. E. El-Hawary, "Emission-Economic Dispatch using a Novel Constraint Handling Particle Swarm Optimization Strategy," in *Electrical and Computer Engineering, 2006. CCECE '06. Canadian Conference on*, 2006, pp. 664-669.

[39] S. Li-quan and G. Xue-yao, "Improved Chaos-Particle Swarm Optimization Algorithm for Geometric Constraint Solving," in *Computer Science and Software Engineering, 2008 International Conference on*, 2008, pp. 992-995.

[40] C. Chun-Hong*, et al.*, "The geometric constraint solving based on memory particle swarm algorithm," in *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*, 2004, pp. 2134-2139 vol.4.

[41] S. Sivasubramani and S. Swarup K, "Multiagent based particle swarm optimization approach to economic dispatch with security constraints," in *Power Systems, 2009. ICPS '09. International Conference on*, 2009, pp. 1-6.

[42] S. Chandrasekaran*, et al.*, "Multi-objective particle swarm optimization algorithm for scheduling in flowshops to minimize makespan, total flowtime and completion time variance," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, 2007, pp. 4012-4018.

[43] H. Chen*, et al.*, "RFID network planning using a multi-swarm optimizer," *Journal of Network and Computer Applications,* vol. In Press, Corrected Proof, 2010.

[44] M. A. Abido, "Multiobjective particle swarm optimization for environmental/economic dispatch problem," *Electric Power Systems Research,* vol. 79, pp. 1105-1113, 2009.

[45] A. C. Briza and P. C. Naval Jr, "Stock trading system based on the multi-objective particle swarm optimization of technical indicators on end-of-day market data," *Applied Soft Computing,* vol. In Press, Corrected Proof, 2010.

[46] C. K. Goh*, et al.*, "A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design," *European Journal of Operational Research,* vol. 202, pp. 42-54, 2010.

[47] B. Alatas and E. Akin, "Multi-objective rule mining using a chaotic particle swarm optimization algorithm," *Knowledge-Based Systems,* vol. 22, pp. 455-460, 2009.

[48] A. B. de Carvalho*, et al.*, "A symbolic fault-prediction model based on multiobjective particle swarm optimization," *Journal of Systems and Software,* vol. 83, pp. 868-882, 2010.

[49] S. Dehuri and S. B. Cho, "Multi-criterion Pareto based particle swarm optimized polynomial neural network for classification: A review and state-of-the-art," *Computer Science Review,* vol. 3, pp. 19-40, 2009.

[50] J. Cai*, et al.*, "A multi-objective chaotic particle swarm optimization for environmental/economic dispatch," *Energy Conversion and Management,* vol. 50, pp. 1318-1325, 2009.

[51] C.-t. Cheng*, et al.*, "Comparison of particle swarm optimization and dynamic programming for large scale hydro unit load dispatch," *Energy Conversion and Management,* vol. 50, pp. 3007-3014, 2009.

[52] S. Z. Zhao*, et al.*, "Dynamic multi-swarm particle swarm optimizer with local search for Large Scale Global Optimization," in *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, 2008, pp. 3845-3852.

[53] Y. Wang*, et al.*, "Trajectory planning for an unmanned ground vehicle group using augmented particle swarm optimization in a dynamic environment," San Antonio, TX, 2009, pp. 4341-4346.

[54] X. Liu*, et al.*, "Particle swarm optimization based on dynamic niche technology with applications to conceptual design," *Advances in Engineering Software,* vol. 38, pp. 668-676, 2007.

[55] W. Du and B. Li, "Multi-strategy ensemble particle swarm optimization for dynamic optimization," *Information Sciences,* vol. 178, pp. 3096-3109, 2008.

[56] X. Yang*, et al.*, "A modified particle swarm optimizer with dynamic adaptation," *Applied Mathematics and Computation,* vol. 189, pp. 1205-1213, 2007.

[57] C. Bae*, et al.*, "Feature selection with Intelligent Dynamic Swarm and Rough Set," *Expert Systems with Applications,* 2010.

[58] C. Ying-Ping*, et al.*, "Particle Swarm Optimization With Recombination and Dynamic Linkage Discovery," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 37, pp. 1460-1470, 2007.

[59] W. Zhenzhen and X. Hancheng, "Dynamic-probabilistic particle swarm synergetic model: A new framework for a more in-depth understanding of particle swarm algorithms," in *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, 2008, pp. 312-321.

[60] J. Zhang*, et al.*, "Particle swarm for the dynamic optimization of biochemical processes," in *Computer Aided Chemical Engineering*. vol. Volume 21, W. Marquardt and C. Pantelides, Eds., ed: Elsevier, 2006, pp. 497-502.

[61] W.-C. Yeh, "A two-stage discrete particle swarm optimization for the problem of multiple multi-level redundancy allocation in series systems," *Expert Systems with Applications,* vol. 36, pp. 9192-9200, 2009.

[62] P.-Y. Yin, "A discrete particle swarm algorithm for optimal polygonal approximation of digital curves," *Journal of Visual Communication and Image Representation,* vol. 15, pp. 241-260, 2004.

[63] A. Unler and A. Murat, "A discrete particle swarm optimization method for feature selection in binary classification problems," *European Journal of Operational Research,* vol. 206, pp. 528-539, 2010.

[64] W.-C. Yeh*, et al.*, "A new hybrid approach for mining breast cancer pattern using discrete particle swarm optimization and statistical method," *Expert Systems with Applications,* vol. 36, pp. 8204-8211, 2009.