Particle Swarm Optimization Methods, Taxonomy and Applications

Davoud Sedighizadeh and Ellips Masehian*

Abstract— **The Particle Swarm Optimization (PSO) algorithm, as one of the latest algorithms inspired from the nature, was introduced in the mid 1990s and since then, it has been utilized as an optimization tool in various applications, ranging from biological and medical applications to computer graphics and music composition. In this paper, following a brief introduction to the PSO algorithm, the chronology of its evolution is presented and all major PSO-based methods are comprehensively surveyed. Next, these methods are studied separately and their important factors and parameters are summarized in a comparative table. In addition, a new taxonomy of PSO-based methods is presented. It is the purpose of this paper is to present an overview of previous and present conditions of the PSO algorithm as well as its opportunities and challenges. Accordingly, the history, various methods, and taxonomy of this algorithm are discussed and its different applications together with an analysis of these applications are evaluated.**

Index Terms – **Heuristic Optimization, Particles Swarm Optimization (PSO), Taxonomy, Applications.**

I. INTRODUCTION

Since conventional computing algorithms are not capable of solving real-world problems because of sometimes having an inflexible structure mainly due to incomplete or noisy data and some multi-dimensional problems, Natural computing paradigms seem to be a suitable replacement in solving such problems. These paradigms consist of simple elements that can solve complicated problems of the real world when working together. It should be mentioned that the main drawback of such paradigms are their indefinite nature and presenting an approximate solution. In general, Natural computing paradigms can be divided into three categories: 1) Epigenesis 2) Phylogeny 3) Ontogeny.

The Epigenesis group is related to a situation in which we would like to develop an intricate structure and to do so, it is necessary to perform a tentative learning. A clear example of this category is Artificial Neural Network (ANN) wherein human's brain is simulated as a complex system.

The phylogeny group is related to EA algorithms. In the algorithms related to this category, there is a competition among agents on survival of the fittest. Algorithms related to this group include Evolutionary Programming (EP), Genetic Programming (GP), and Differential Evolutionary (DE).

The Ontogeny group is associated with the algorithms in which the adaptation of a special organism to its environment is happened. The algorithms like PSO and Genetic Algorithms (GA) are of this type and in fact, they have a cooperative nature in comparison with other types [16]. The advantages of above-mentioned categories can be noted as their ability to be developed for various applications and not needing the previous knowledge of the problem space. Their drawbacks include no guarantee in finding an optimum solution and high computational costs in completing Fitness Function (F.F.) in intensive iterations. Among the aforementioned paradigms, the PSO algorithm seems to be an attractive one to study since it has a simple but efficient nature added to being novel. It can even be a substitution for other basic and important evolutionary algorithms.

The most important similarity between these paradigms and the GA is in having the seam interactive population. This algorithm, compared to GA, has a faster speed in finding the solutions close to the optimum and it is faster than GA in premature convergence [4].

II. DESCRIPTION OF PSO

Kennedy and Eberhart [31], considering the behavior of swarms in the nature, such as birds, fish, etc. developed the PSO algorithm. The PSO has particles driven from natural swarms with communications based on evolutionary computations. PSO combines self-experiences with social experiences. In this Algorithm, a candidate solution is presented as a particle. It uses a collection of flying particles (changing solutions) in a search area (current and possible solutions) as well as the movement towards a promising area in order to get to a global optimum.

$$\begin{aligned} &prtvel_j^i = c(w \times prtvel_j^{i-1} + \\ &c_1 \times rand \times (pbest_j^{i-1} - prtpos_j^{i-1})) + , \\ &c_2 \times rand \times (gbest^{i-1} - prtpos_j^{i-1}) \end{aligned} \tag{1}$$

Manuscript received 18 JUNE. This work was supported in part by the Tarbiat Modares University under Grant 15791-5-12-87.

D. Sedighizadeh is with the Faculty of Engineering, Tarbiat Modares University, Tehran, 14115-143, Iran (phone: +98-21-82883145; fax: +98-21-82884180; e-mail: sedighizadeh@modares.ac.ir).

*Corresponding aurhor: E. Masehian, is an assistance professor in the faculty of engineering, Tarbiat Modares University, Tehran, Iran（e-mail: Masehian@modares.ac.ir.）

$$c = \frac{2}{\left|2 - j - \sqrt{j^2 - 4j}\right|}, \; j = j_1 + j_2, \; j > 4 \quad (2)$$

$$prtpos_j^i = prtpos_j^{i-1} + prtvel_j^i \quad (3)$$

Where,

c_1, c_2 : The balance factors between the effect of self-knowledge and social knowledge in moving the particle towards the target. Usually the value 2 is suggested for both factors in the literature)

rand :A random number between 0 and 1, and different at each iteration

w : Inertia weight

$pbest_j^i$: The best position of a particle

$gbest^{i-1}$: The best position within the swarm

$prtvel_j^i$: The velocity of jth particle in ith iteration

$prtpos_j^i$: The position of jth particle in ith iteration

Since, in the above algorithm, there is the possibility of particles movement to out of the problem space [15], an upper velocity bound for particle movement is specified.

One of the PSO problems is its tendency to a fast and premature convergence in mid optimum points. A lot of effort has been made so far to solve this problem. For instance, in [14] the best value for w in (1) is set to 0.9, which linearly decreases to 0.4. Moreover, χ (i.e., contraction parameter) reduces the necessity of haltering velocity.

III. Major PSO-based Algorithms

2-D Otsu PSO (TOPSO) – This algorithm is a combination of the PSO and the optimal threshold selecting search in order to improve the PSO performance [87].

Active Target PSO (APSO) – In this algorithm, in addition to two existing terms; namely the best position and the best previous position for particle velocity updating, a third term called 'Active target' is also utilized. Calculating the third term is complicated and it does not belong to the existing positions. This method maintains the diversity of the PSO as well as not trapping in the local optimum [103].

Adaptive PSO (APSO) – During the running process of the PSO, sometimes a number of particles are inactive, that is, they do not have the ability of local and global searching and do not change their positions a lot, so their velocity is nearly reached to zero. One solution is to adaptively replace the current inactive particles with fresh particles in a way that the existing PSO-based relationships among the particles are kept. This is done by APSO method [89].

Adaptive Mutation PSO (AMPSO) –This algorithm utilizes the adaptive mutation using Beta distribution in the PSO. It includes two types: AMPSO1 and AMPSO2. The former mutes the best individual position in the swarm and the latter mutes the best global position [62].

Adaptive PSO Guided by Acceleration Information (AGPSO) – This algorithm is for improving the PSO efficiency in finding the global optimum. The acceleration item is also added to position and velocity updating equations and then, convergence analysis is performed [100].

Angle Modulated PSO (AMPSO) – This algorithm employs a trigonometry function to generate a bit string. Its difference with the Binary PSO (BPSO) algorithm lies in its high computational efficiency. That is, it avoids the generation of high-dimensional binary vector and thus, its discretion process is not complicated. Moreover, it changes all of the high-dimensional problems to a four-dimensional problem. Hence, it saves a large amount of the memory and is easy to run [60].

Attractive Repulsive Particle Swarm Optimization (ARPSO) – This algorithm has been developed to remove the PSO's drawback in premature convergence. It includes an attractive phase and a repulsive phase. In the attractive phase, the addition operator is used among the equation terms for velocity updating. In the repulsive phase, the subtraction operator is employed. Indeed, the particles are attracted towards each other in the attractive phase and get away from each other in the repulsive phase [68].

Augmented Lagrangian PSO (ALPSO) – This algorithm is a combination of Augmented Lagrangian method and the PSO algorithm. It is applied to optimization problems having equal and unequal constraints [73].

Best Rotation PSO (BRPSO) – This algorithm is used to optimize multimodal functions and in fact, the swarm is divided into several sub-swarms. It is worth mentioning that the swarm separation and its division on several populations do not look reasonable for single modal problems. However in normal PSO in multimodal functions the wide knowledge of the whole population performance make the system converges too fast and also increases the probability of stagnation into local minima but in BRPSO when best rotation is executed, stagnation on local minima is avoided by forcing populations to move from one local minimum to another one, increasing the exploration of the problem space between different local minima. This algorithm is in a way that a periodically rotation is performed among the particles of different sub-swarms [2].

Binary PSO (BPSO) – The difference between PSO and BPSO lies in their defined searching spaces. In the typical PSO, moving in the space means a change in the value of position coordinates in one or more of existing dimensions. However, in the BPSO moving in the spaces means a change in the probability of the fact that the value of position coordinate is zero or one [32].

Co-evolutionary PSO – This algorithm was proposed in 2002 in [77].

Combinatorial PSO (CPSO) – This algorithm is employed to optimize hybrid problems (consisted of continuous and integer variables) [26].

Comprehensive Learning PSO (CLPSO) – In [43], the new velocity updating function is proposed and employed to construct CLPSO and then the new algorithm is tested using a group of benchmark functions.

Concurrent PSO (CONPSO) – In 2004, the CONPSO algorithm was developed in [5].

Constrained optimization via PSO (COPSO) – The COPSO algorithm is applied to constrained single-objective problems. In this algorithm, a technique is employed to investigate the constraints and it has an external file, called "Tolerant", to save the particles. Indeed, in this technique some particles are missed through setting constraints. In order to develop the lifetime of these particles, the above-mentioned external file is utilized and a ring topology structure is employed. In fact, the COPSO is a kind of improvement in Lbest version of the PSO. Moreover, the external procedure, which maintains swarm diversity and guidance towards good points keeping the self-setting capacity, are utilized [1].

Cooperative PSO (CPSO_M) – In 2004, this algorithm was presented in [56] wherein a multi-cooperative algorithm was schemed.

Cooperative PSO (CPSO_S) – In 2004, this algorithm was schemed in [81] in which a single cooperative algorithm was introduced.

Cooperatively Coevolving Particle Swarms (CCPSO) – This algorithm is suitable for large-scale problems. It breaks the problem into some smaller-scaled ones in a way that the internal dependencies of generated particles are in the possible least values. Then, these particles will become cooperated [95].

Cooperative Multiple PSO (CMPSO) – Since the PSO efficiency when solving multi-dimensional problems is reduced, the CMPSO algorithm is introduced to overcome this problem. This algorithm has all conductivity and control properties of the PSO [17].

Cultural based PSO (CBPSO) – This algorithm is in fact the use of PSO in cultural algorithm (CA) framework. Because of PSO's drawback in finding the global optimum and on the other hand, the effecting of the CA in finding the global optimum due to having multiple evolutions and multiple progresses, using them simultaneously can enhance the PSO [29].

Dissipative PSO (DPSO) – Sometimes the evaluation in the PSO becomes static because of swarm’s tendency to get the equilibrium status. Thus, the algorithm will be prevented from searching for more areas and it may occasionally be trapped in a local minimum. In order to overcome this problem, a dissipative system is made using the DPSO algorithm introducing the negative entropy and producing craziness among particles. Utilizing of this system will practically prevent the above-mentioned stagnancy [90].

Divided range PSO (DRPSO) – In this method, wherein there are several objective functions, particles are first divided to sub-swarms based on one of the objective functions value. Next, the discrete PSO algorithm is run in each sub-swarm. If the stop condition is satisfied, the algorithm will finish; otherwise, the particles are gathered again and are ordered based on the next objective function and the categorizing takes place once more. This algorithm is employed for the clustering of hoc and mobile networks [27].

Dual Similar PSO Algorithm (DSPSOA) – This algorithm is schemed in [42] wherein through the improvement of the option modes of gbest and pbest of the PSO algorithm, an effective dual similar particle swarm optimization algorithm (DSPSOA) is presented.

Dynamic adaptive dissipative PSO (ADPSO) – In this algorithm, on the one hand, a dissipative is made for the PSO introducing negative entropy and on the other hand, a mutation operator is utilized to increase the variety in the swarm when it reaches an equilibrium condition in last runs. Thus, it generates an adaptive strategy for inertia weight in order to keep the balance between the local and global optimality [75].

Dynamic and Adjustable PSO (DAPSO) – In order to make a balance between the discovery and extraction in the PSO and also to keep and protect the particles diversity, DAPSO algorithm has been proposed in which the distance of each particle to the best position is calculated to adjust the velocity of particles in each step [44].

Dynamic Double Particle Swarm Optimizer (DDPSO) – This algorithm, using a convergence analysis, guarantees the convergence to the global optimal solution. Particle position constraints are set dynamically in this method [12].

Dual Layered PSO (DLPSO) – The DLPSO algorithm is developed to design a neural network. This algorithm optimizes the network in an architectural layer. It is used for neural network joint weights. A classic boost power transformer is employed to test neural network controllers [78].

Dynamic neighborhood PSO (DNPSO) – The DNPSO method has some modifications to the conventional PSO. In this method, instead of using the current Gbest in the PSO, another parameter, called Nbest, is utilized. This term is the best particle among the current particle’s neighbors in a specified neighborhood. This method discusses that the selection of neighbors for the current particle, as an objective, is multi-objective. In addition, the selection of their best is another objective [22].

Estimation of Distribution PSO (EDPSO) – This algorithm is a hybrid of the PSO and Estimation of Distribution Algorithm (EDA). Indeed, the ED algorithms—using the obtained information from stochastic models upon which good solution areas on distribution are generated during the optimization process—try to find better areas. This feature of such an algorithm is utilized to improve the performance of PSO [35].

Evolutionary Iteration PSO (EIPSO) – This algorithm is a combination of the PSO and Evolutionary Programming (EP). Thus, it is able to increase the computational efficiency of EP and it can avoid trapping the algorithm in local optimum [38].

Evolutionary Programming and PSO (EPPSO) – This algorithm is a combination of the PSO and EP. Indeed, the combination of these two algorithms will cause a help for the PSO capability in making a balance between local and global search to the faster convergence of the EP algorithm. On the other hand, the PSO's drawback in lacking diversity among the particles with mutation between elements in the EP is to some extent removed [86].

Extended Particle Swarms (XPSO) – Using the Genetic Programming, various algorithms driven from the PSO can be obtained in [30].

Extended PSO (EPSO) – In this algorithm, the contemporary advantages of Gbest and Lbest versions are utilized. In fact, a hybrid of both is employed in velocity updating equation. The difference between these algorithms with the Fully-informed PSO (FIPS) algorithms lies in less computational costs [67].

Fitness-to-Distance Ratio PSO (FDRPSO) – In 2003, the FDROSO was represented in [65]. The new proposed algorithm moved particles towards nearby particles of higher fitness instead of attracting each particle towards just the best position discovered so far by any particle. This was accomplished by using the ratio of the relative fitness and the distance of other particles to determine the direction in which each component of the particle position needed to be changed.

Fully-informed PSO (FIPS) – In this algorithm, which mainly differs in topology type of particles, all particles have an information source and there is no difference in the amount of their information [52].

Fuzzy PSO (FPSO) – In the FPSO methodology, the idea of PSO is used together with an explicit selection procedure. Moreover, self-adapting characteristics are utilized to set the parameters. Generally, the replication, mutation, reproduction, evaluation and selection operations are employed in this algorithm [76].

Gaussian PSO (GPSO) – PSO conjures an image of particles searching for optimal ways that bees buzz around flowers. One approach at visualizing the swarm graphs, where all the particles are each generation, thus demonstrating the random nature associated with swarms of insects. Another approach is to show successive bests, thus showing the way that the swarm makes progress. Some have even looked at the specific search path of the particle that eventually finds the optimums. These approaches provide limited understanding of PSO. This approach presents a new visualization approach based on the probability distribution of the swarm, thus the random nature of PSO is properly visualized. The visualization allows better understanding of how to tune the algorithm and depicts the weaknesses. A new algorithm based on moving the swarm a Gaussian distance from the global and local best is presented in [72].

Geometric PSO (GPSO) – This algorithm uses a geometric framework for connection between the PSO and evolutionary algorithms and by doing this, the generated algorithm will be able to be applied to both continuous and combinational spaces and it will cover most of the problems [53].

Genetic PSO (GPSO) – In 2006, GPSO was derived from the original PSO. It was incorporated with the genetic reproduction mechanisms, namely crossover and mutation. [96].

Genetic binary PSO model (GBPSO) – This algorithm was developed to increase the dynamic conditions and discovery power in the swarm. In the BPSO, bear and death parameters are employed. In other words, according to BPSO principles, the positions and velocities are updated and then, some of the child particles are added to swarm and some others die and are separated from the swarm. It is worth nothing that in binary state each particle is considered as a chromosome and chain with the size of space dimension [70].

Greedy PSO (GPSO) / DS-BPSO (Double-Structure coding Binary PSO) – This algorithm was first schemed to solve the knapsack problem which was very successful. It made use of Greedy transform method. This algorithm is a hybrid evolutionary algorithm that combines the binary PSO with the Greedy transform. In addition, binary PSO with double-structure coding is also introduced which is in fact the use of BPSO for 3-SAT problems [36].

Gregarious PSO (GPSO) – In this algorithm, particles only use social knowledge for discovery in the search space. If they are trapped in the local optimum, a stochastic velocity vector is employed. In these algorithms, unlike the PSO, the last integration is used in resulting the parameters and this is a sort of self-setting for the parameters [64].

Heuristic PSO (HPSO) – In 2007, a variant of particle swarm optimizer called HPSO was introduced, which differed from the original PSO in choosing the next particle to update its velocity and position. The utilized approach in this algorithm can speed up the convergence rate of the swarm to a local optimum. To avoid premature convergence, particles' positions are re-initialized randomly when their position is close to the global best position. The combination of the heuristic updating and the position re-initialization helps HPSO outperform the basic PSO and some variations of PSO in some test cases [36].

Hierarchical PSO (HPSO) – The HPSO was presented in 2004 in [25]. In this method, the particles are arranged in a dynamic hierarchy used to define a neighborhood structure. Depending on the quality of their so-far best-found solution, the particles move up or down the hierarchy. Furthermore, another algorithm called 'Partitioned Hieratical PSO (PH-PSO) was presented by them in 2004 [23].

Hierarchical recursive-based PSO (HRPSO) – Self-generation fuzzy modeling systems through HRPSO are launched in [18].

Hybrid discrete PSO algorithm (HDPSO) – This algorithm is utilized for scheduling the flow shop system and in fact, each particle indicates a job sequence as a solution [9].

Hybrid gradient descent PSO (HGPSO) – In this method, two movement terms—one towards the global optimum and the other in negative direction of gradient—are utilized. Although moving in negative direction of gradient may end in a local optimum, the term of moving towards the global optimum can prevent it. Gradient calculation will increase the computational efforts but in return, calculations of particle neighbors are ignored [57].

Hybrid PSO with simulated annealing (SAPSO) – The PSO is very efficient in finding the global optimum but on the other hand, it may be trapped in local optimum. The GA, in return, is very valid in finding the local optimum and can avoid trapping in local optimum, but it is weak in finding the global optimum. Thus, the hybrid of these two algorithms can cover their drawbacks. In order to hybrid these two approaches, simulated annealing is performed on each particle after their random production then the position and velocity updating is done on them. This procedure is continued until 'stop' instruction is given [85].

Hybrid Recursive PSO learning algorithm (HRPSO) – This algorithm—which is a combination of C-mean fuzzy clustering, the PSO algorithm and recursive least, and squares (RLS)—was introduced to design a RBF neural network in order to quick estimation of two complexes and nonlinear function [11].

Hybrid Taguchi PSO (HTPSO) – This algorithm combines the PSO with the Taguchi selection method. In this approach, the intelligent particles are selected [69].

Interactive PSO (IPSO) – In IEC, user's idea replaces the F.F. That is, the user gives an idea based on current criteria in each case. Since in the PSO, despite EC and IEC, information transmission does not solely take place among the particles of that iteration, the IPSO and IEC mechanisms are different from each other. The IPSO is the same as PSO procedure and the difference is that the best particle determination is done by the user and not by utilizing the F.F [49].

Immune PSO (IPSO) – The IPSO algorithm makes use of the PSO advantages to improve the mutation mechanism in the immune algorithm. Evidently, in a few other cases, some information is used as immune operator in the PSO [45].

Improved Particle swarm optimizer (IPSO) – This algorithm is based on PSOPC. Moreover, it uses a harmony search. It utilize a mechanism call fly-back in order to employ the constraints [104].

Iteration PSO (IPSO) – The IPSO is introduced in [37]. In IPSO, a new index called iteration best is incorporated into the PSO to improve solution quality and computation efficiency. Expanding line construction cost, contract recovery cost, demand contract capacity cost, and penalty bill are considered in selecting the optimal contract capacities.

Map Reduce PSO (MRPSO) – When solving problems with large data values, the PSO may not have the necessary efficiency since the evaluation of individual functions can be time consuming. Hence, the MRPSO algorithm has been proposed, which is in fact a parallel run of the PSO for computationally compressed functions [50].

Modified Binary PSO (MBPSO) – In this algorithm, which is a modified version of the BPSO algorithm, all of the particles are produced as binary vectors and in a random way. Then, the least value of position is used to map the binary space to the permutation space. In this algorithm, new equations are employed to update the position and velocity [97].

Modified Genetic PSO (MGPSO) –This algorithm is in fact the combination of the two GPSO and DE (differential Evolution) algorithms. In this algorithm, it is tried to improve the GPSO performance. Updating of the next position is done by both algorithms for each particle and the better result will be the benchmark for the next movement of the particle [105].

Multi-Grouped PSO (MGPSO) – The MGPSO is presented in [74] to solve multi-modal problems. In this method, the swarm is divided to several groups based on similarity.

Multi-Objective PSO (MOPSO) –The MOPSO was introduced in 2007 when the CMOPSO and Hyper Volume-based MOPSO (HMOPSO) were also presented [54].

Nbest PSO – The Systems of unconstrained equations using Nbest PSO is solved in [7].

Neural PSO (NPSO) – In this method, wherein a feed forward neural network is combined with the PSO, neural particles are defined in space like feed forward neural network. In this neural network, the learning process is the movement of particles following the bests in space [13].

New optimization algorithm based on PSO (NPSO) – In this method, the new contribution relates to the introduction of a new "momentum term" which is known to influence the convergence properties of the original PSO algorithm. It is shown that the new algorithm structure, called NPSO, can solve the problem of premature convergence—widely experienced in the original PSO algorithm—and also can make the particles' optimal search process "truly" adaptive [101].

NewPSO (NPSO) – In this algorithm, the worst are used instead of bests to calculate the Pbest and Gbest. These terms, however, are utilized with negative sign in velocity updating equation. In other words, using this process it is tried to get farther from the worst instead of getting closer to the bests [93].

Niching PSO – This algorithm is employed for multiple optimizations. In this algorithm, the traditional approach is run first. During the run time of the PSO, particles are monitored individually. The particle, whose fitness in each iteration changes a little or does not change at all, is separated from the swarm and a sub-swarm is formed. As the algorithm goes on, the members of main swarm are reduced and new sub-swarms are created. In fact, these created sub-swarms are dynamically for finding all of the global and local optimums in a parallel and simultaneous way [8].

Novel Hybrid PSO (NHPSO) – This algorithm is for making the PSO more efficient in solving high-dimensional problems. It is a hybrid of the PSO and the harmony search scheme. Indeed, the harmony search scheme helps the better searching of the PSO and this will cause the PSO to be more enhanced in exploitation [39].

Optimized PSO (OPSO) – This algorithm has swarms within a swarm to optimize the free parameters of the PSO. Test results reveal the better performance of this method compared to other methods [51].

Orthogonal PSO (OPSO) – In this algorithm, in order to update the velocity, a system called Intelligent Move Method (IMM) is used instead of the conventional system. In fact, in the IMM strategy, the divide and conquer approach is utilized to determine the next move of the particle. The results reveal that this algorithm performs better than PSO in optimization problems with large-scale parameters [21].

Parallel PSO (PPSO) – In 2005, the Parallel PSO algorithm was introduced in [10]. In this algorithm, time requirements for solving complex large-scale engineering problems can be substantially reduced using parallel computation. Motivated by a computationally demanding biomechanical system identification problem, a parallel implementation of a stochastic population-based global optimizer—the Particle Swarm Algorithm—is introduced as a means of obtaining increased computational throughput. The Particle Swarm requires very few algorithmic parameters to define convergence behavior due to its simplicity and as a population-based optimization method; it is a natural candidate for concurrent computation.

Parallel Asynchronous PSO (PAPSO) – This algorithm was extracted from the PSPSO algorithm. In the PAPSO, the difference between being synchronous and asynchronous lies in the position and velocity updating equations. Particles and velocity updating is done continuously based on existing and accessible information. This algorithm generates a dynamic view of load balancing along with a chain-duty central approach to reduce the imbalance load [33].

Parallel synchronous PSO (PSPSO) – This algorithm performs the position and velocity updating at the end of each iteration using wholly simulating. It uses a constant load balancing in which the assigned task to each processor in the total time is determined [28].

Parallel vector-based particle swarm optimizer (PVPSO) – This algorithm has employed several algorithms including PSO, Niche PSO, and Vector-based PSO. It tries to optimize the existing sub-swarms in the niches simultaneously. It is also tried to converge the total swarm to optimality all at the same time. Thus, this approach is applied to problems related to simultaneous optimization of several functions [71].

Perturbation PSO (PPSO) – This algorithm, trying to remove the PSO drawback in trapping in local optimum, avoids trapping in local optimum through making perturbation in static particle. That is, it changes the position and velocity updating equations but keeps the existing equations in the PSO for the others [98].

PSO-bounds – In 2008, this algorithm was introduced and presented in [16]. For the PSO Bounds swarm, the model is a vector containing the lower bound, the higher bound, and the probability of the value existing in the higher half for all of the dimensions.

PSO with area extension (AEPSO) – This algorithm was designed for movement of several robots in an area. It has in fact some modifications to the conventional PSO. These modifications are with information increasing from an extended area. In order to get this goal, a series of heuristics are utilized to update the particles velocity. Moreover, some heuristics are employed to avoid trapping in local optimum and also to prevent the problem from being stuck [3].

PSO with behavior of distance (BDPSO) – In this algorithm the flying area of particle is divided into various areas. Consequently, the swarm will not have a constant behavior and it will have a different behavior depending on which area it is flying. That is, in attraction area particles fly faster towards the best position and in repulsive area they move at a normal rate [23].

PSO with craziness and hill climbing (CPSO) – In most of the algorithms related to the PSO, it is tried to make a balance between discovery and extract in the algorithm. This balancing will be an important success. The CPSO algorithm, for optimizing multimodal functions, uses the craziness and hill climber to enhance the discovery and extract, respectively [59].

PSO with Escape Velocity (EVPSO) – This algorithm is to avoid quick convergence and furthermore, it is for increasing the variety in swarm. In fact, an escape velocity is added to all of the particles so that the mentioned objectives are obtained [84].

PSO with passive congregation (PSOPC) – Grouping has two types: 1) Aggregation: which itself has two kinds. The first kind is passive aggregation in which a passive group is with a physical process; like planktons swarm floating on the water that the water flow keeps them together. The second kind is active aggregation in which the aggregation is performed by an absorbent source. The absorbent source may be food or water. 2) Congregation: This is different from aggregation. That is, the absorbent supply—and not external and physical factors—is the group forced by it. This type is also divided in two kinds: 1) Passive congregation in which there is an attraction from one particle to others but is not shown a social behavior. 2) Social congregation in which there is a social behavior among the particles and they are strongly related to each other. Since in some groups there may be a selfish behavior in information sharing, e.g. fish school, a selfish behavior may lead in forming a passive group. A passive swarm model can be added to the PSO in order to increase its efficiency, which is the PSOPC. This term is the same as randomly selected particle from the current swarm and particle [19].

PSO with spatial particle extension (SEPSO) – In this approach, similar particles are collected in a sub-swarm called species. The Euclidean distance is used as the criterion of particles similarity. In this method, particles move in their species and in fact a parallel multi-objective optimization happens. The amount of convergence is increased with running the PSO in each species and their continuous reconstruction [34].

Predator Prey PSO *(PPPSO)* – In the PPPSO, there are definitions including predator and prey. In this algorithm, predators follow the prey and preys escape from predators. This is done to avoid local optimum and to move towards the global optimum [24].

Principal Component PSO (PCPSO) – The PCPSO is employed to reduce the time complexity of the problem in high dimensions. In this algorithm, particles are flown in an n-dimensional space and contemporarily, they are flown in an m-dimensional space (m is less than n). That is, the particles are flown simultaneously in two separate spaces [82].

Pursuit-Escape PSO (PEPSO) – This algorithm uses the idea of small fish behavior and predator whales. That is, in the contrast between those two groups, small fish are escaping and the whales are per suiting. In simulating this behavior in the PSO, the swarm is in fact divided into two groups namely, escaping swarm and per suiting swarm. This concept will make intensification—resulting from per suiting group—and diversification resulting from the escaping group. This will avoid trapping in local optimum. Furthermore, a suitable balance is generated between diversification (global search) and intensification (local search) [20].

Quantum Delta- Potential-Well-Based PSO (QDPSO) – In the PSO, a particle is presented by its path which is in fact position and velocity values. But in quantum that is based on uncertainty principle the path does not make sense because of the fact that position and velocity values can not be determined simultaneously. Thus, definitions are different in quantum; though the PSO principle is maintained. In quantum, an algorithm called: ‘Quantum Delta-Potential-well- based PSO is employed to solve the problems [79].

Quantum-inspired version of the PSO algorithm (QPSO) – The QPSO algorithm permits all particles to have a quantum behavior instead of the classical Newtonian dynamics assumed so far in all versions of the PSO. Hence, instead of the Newtonian random walk, some sort of “quantum motion” is imposed in the search process. When the QPSO is tested against a set of benchmarking functions, it demonstrates superior performance as compared to the classical PSO but under the condition of large population sizes. One of the most attractive features of the new algorithm is the reduced number of control parameters. Strictly speaking, there is only one parameter required to be tuned in the QPSO [79].

Quadratic Interpolation PSO (QLPSO) – This algorithm utilizes a multi-parent, quadratic crossover/reproduction operator. The reproduction operator, in fact, has been loaned from EA. This algorithm uses the idea of having several partners. In this algorithm, the swarm leader is determined in each iteration. Then, its partners are selected among other particles. Next, employing a crossover operation called "quadratic crossover", off spring production is performed. The new particle is accepted into the swarm when it is better than the best particle existing in the swarm [61].

Restricted Velocity PSO (RVPSO) – The PSO algorithm is for unconstrained optimization problems. That is, its search mechanism is a type which the search space is infinite. However, there are sometimes problems in which the search space has an acceptable range. In order to be able to solve such problems, the RVPSO approach is applied. In this approach, the particle velocity is constricted considering the constraint [48].

Self-adaptive Velocity PSO (SAVPSO) – Since the PSO is inherently for unconstrained problems, many challenges have been arisen to make a mechanism in the algorithm in order to handle the generated constraints (or knowledge about the feasible region). The SAVPSO is an algorithm in which a mechanism is employed to investigate the impact of constraints in the PSO algorithm [47].

Self-organization PSO (SOPSO) – In this algorithm, in addition to particle information and total swarm information, a feedback agent is employed to improve the particle performance. Indeed, the particle, utilizing the feedback information of total swarm, sets and improves its behavior in next iteration. Generally, this agent will lead in improvements in discovery and extract of the particles. Moreover, it causes an incensement in the variety among the particles. The main objective of this algorithm is to avoid premature convergence of the total algorithm [28].

Sequential PSO (SeqPSO) – The SeqPSO algorithm was introduced in [51] wherein a sequential approach was employed. [102].

Set PSO – In 2006, an algorithm called Set PSO was introduced in [55].

Shuffled Sub-Swarms Particle Optimizer (SSPSO) – This algorithm is performed to enhance the diversity of particles in order to improve the performance of the PSO [83].

Species in a Particle Swarm Optimizer (SPSO) – In this method, similar particles are categorized in different sub-swarms. The similarity criterion, for instance, may be the Euclidean distance. Each of these sub-swarms is called a species. In each species, the particle having the best fitness will be species seed and will have a species radius. In fact, each particle evolves inside its species and a multiple parallel evaluation takes place. Hence, this method is applied to multi-objective optimization problems [41].

Swarm-Music – In 2003, the Swarm-music algorithm was presented in [6].

Trained PSO (TPSO) – This algorithm using a mechanism tries to reduce the completion complexity and convergence time. It is applied to Ad-Hoc communication networks. In fact, the particles in this network are moved. In ordinary states, moving these particles will increase the computation complexity and thus, traffic will reduce the convergence time. These drawbacks will be removed by training the particles [80].

Two-Swarm-based PSO (TSPSO) – The aim of this algorithm is to escape from being trapped in local optimum and to avoid quick convergence. Two swarms with different parameters are flown is the space. That is, the particles of both swarms have different paths from each other. One of them will enhance the capability of finding the global optimum and the other, using the Roulette-wheel-selection based stochastic selection scheme, will enhance the local discovery [40].

Unconstrained PSO (UPSO) – The PSO is divided to constrained and unconstrained categories depending on the limits of its velocity or position parameter. The algorithm is generally constrained, as is the classic form. It is worth mentioning that the position and velocity updating equations are the same in both states. However, in the classic mode (constrained), there are up and down constraints for position and velocity where if they are exceeded, these constraints will be considered. But in UPSO mode such a constraint does not exist [99].

Unified PSO (UPSO) – This algorithm is in order to simultaneously utilize the advantages of both Gbest and Lbest approaches in the PSO. In fact, the velocity updating equation is divided in two parts that each part calculates the velocity based on information type (Gbest and Lbest). An equivalency factor is also employed [63].

Variable Neighborhood PSO (VNPSO) – This algorithm is a Meta heuristic algorithm and is a hybrid of the PSO and variable neighborhood search (VNS). It is applied to flexible multi-objective job-shop scheduling problems. In this algorithm, the VNS is employed for local optimizer. This search scheme escapes from trapping in local minimum and performs it by repeated searches from start point to a local optimum which is better than the current existing local optimum [46].

Vector Evaluated PSO (VEPSO) – This algorithm is a novel algorithm based on a multi-objective interaction sort of the PSO [58]. VEPSO is a co-evolutionary multi-objective variant of the popular PSO.

Velocity Limited PSO (VLPSO) – If particles' moving velocity is limited in various ranges, different optimal solutions can be obtained. Thus, considering and regarding up and down constraints for the velocity and position, the VLPSO approach is proposed. The strategy of this approach is in a way that only the particles satisfying the constraints will be kept and the others are eliminated [91].

Velocity Mutation PSO (VMPSO) – This algorithm is a hybrid algorithm derived from the PSO, which has been presented to system structure identification [92].

Vertical PSO (VPSO) – Since in some of the PSO iterations the global optimum point of the algorithm is not improved and the particle remains close to the optimum point, and as the next movement of the particle highly depends on its previous movement, moving in that path is done very difficultly and it moves towards a local optimum. Hence, the

VPSO algorithm has been developed in which the particles fly in both global and vertical directions and the existing problem is to some extent removed [94].

The main parameters used in some of the methods ramified from the original PSO algorithm are described more detailed in Table I as a guide for parameter setting and utilization.

IV. PSO'S TAXONOMY

In this section, the assessment of taxonomy in the PSO algorithm is presented. A PSO’s Taxonomy is shown as Fig. I and the major PSO bibliographies are illustrated as Table II.

Continuity – From the viewpoint of the continuity in the space where the particles are located, the PSO is divided into two parts; namely, continues and discrete. In the continuous state, particle's movement path is as a change in particle's positions in same dimensions. However, in the discrete state, this movement path is as a change in the probability of the fact that the value of position coordinate is zero or one.

Fuzzified – The PSO is investigated through two views from Fuzzified point of view. In some applications of the PSO, like multi-objective quadratic assignment problems, the fuzzy mode of the algorithm is assessed. That is, the presentation of velocity and position in vector form is changed from real vectors to fuzzy matrixes.

Accordance – Sometimes, during the runtime of the PSO, swarm evolution process is nearly stopped and becomes stationary. This is occasionally because of the fact that some particles become inactive, that is, they are unable in local and global searching; hence, they do not change much to their pervious positions and their velocity is near to zero. One solution is to adaptively replace these inactive particles with fresh particles in a way that the existing rotations among the particles, which are based on PSO Particles, are maintained. This is done by APSO method. But sometimes this stop is due to swarm tendency in getting the equilibrium state which prevents searching for more areas and it may be trapped in a local minimum. In order to solve this problem, using the Dissipative PSO algorithm (DPSO), a dissipative system introducing negative entropy and making chaos among the particles is generated. By utilizing this system, it is practically prevented from the above-mentioned stationary state. Here, the two previously mentioned approaches are employed simultaneously. That is, on the one hand, a dissipative is developed for the PSO introducing negative entropy, and on the other hand, a mutation operator is used to increase the swarm variety when the algorithm reaches an equilibrium state in last runs. Therefore, an adaptive strategy is developed for inertia weight updating in order to keep the balance between local and global optimizing. This has been performed in the Dynamic Adaptive Dissipative PSO (ADPSO) algorithm. Obviously, the printed concepts can be assessed both in a static or a dynamic environment.

Attraction – In order to solve problems such as premature convergence, there are three approaches, namely attraction, repulsive, and attraction/repulsive. In the attraction phase, the addition operator is employed to update the velocity equations and in the repulsive phase, the subtraction operator is utilized. Indeed, the particles are attracted to each other in the attraction phase and they get away from each other in the repulsive phase. In the attraction/repulsive state, the swarm evolution is performed through both attraction and repulsive phases.

Topology – the PSO algorithm is divided into various topologies from the viewpoint of accessibility of particles information. In Gbest type, all of the particles are related with each other. In fact, all of the particles are affected from each other. But in the Lbest topology, each particle is related with neighbor particles and a looped network is formed. Another topology is the pyramid which is like a three dimensional triangle that shows the relation between the particles in a three-dimensional way. In the Star topology a central node influences the whole population and is affected by it, too. The Small topology is a graph made up of isolated sub-swarms and particles and it is in fact an instance of being heterogeneous. In the Von-Neumann topology, the up/down and each side neighbors are located on a loop in a two-dimensional space. The Vis-Best topology, which is introduced for the first time in this paper, is in fact an average state of prevalent Lbest and Gbest topology. In this topology, the information can be divided not only among the particles of instantaneous neighbor in the discrete state, but also among all of the particles in the observation region of a particle. Indeed, the particles which are in the observation region of each other are aware of each other's best position and this can draw a condition closer to fact. In addition to the previously mentioned topologies, there also exist other topologies, which are created randomly.

Activity – Activity has two types. In active state, there is an attraction from each particle for other swarm in a way that a social behavior is expressed in whole swarm. But in passive state, even though there is an attraction for each particle from other particles, a social behavior is not shown in the whole swarm.

Grouping – Grouping has two kinds: 1) Aggregation, which is itself divided into two categories; the first type is inactive (passive) aggregation in which there is a passive swarm with a physical process; like planktons floating on the water that the water flow has kept them together. The second type is the active aggregation in which the aggregation is performed by an absorbent source. This source can be food or water. 2) Congregation, which is different from aggregation. That is, the absorbent source is the group forced by itself and there are no external and physical factors. It is divided into two types as well: 1) passive type in which there is an attraction from one particle to other particles, but a social behavior is not expressed. 2) Social type in which there is a social behavior among the particles and they are related severely with each other.

Mobility – In order to increase the PSO's efficiency, sometimes with a dynamic view and employing dynamic mechanisms, it is tried to update the particles positions. For instance, in order to make a balance between exploitation and exploration in the PSO and keeping the diversity of the particles, the DAPSO algorithm has been proposed in which each particle's distance to the best position is calculated to adjust the velocity of the particles. But in contrast, traditional static mechanisms are utilized.

Divisibility – From the viewpoint of particles divisibility, the PSO is divided into divisible and non-divisible types.

Sometimes, in other to increase the algorithm efficiency or to increase the variety in the swarm or its multi-objectiveness, the main swarm is divided into sub-swarms.

Types of Particles – Sometimes, the particles in the PSO are allowed to follow quantum behavior instead of following the classic dynamic of Newton. In other words, the particles use a quantum movement in the search space instead of Newton movement. In high dimensions, the results are better than the classic state. This is especially seen in reduction of needed parameters for setting.

Interaction – In IEC, F.F. is replaced with user's idea. That is, the user gives opinion about each particle considering the existing criteria. Since in the PSO, unlike EC and IEC, data transmission is not done just among the particles of that iteration and in fact, particle information of pervious iterations is also applied, the IPSO and IEC mechanism are different from each other. The IPSO is the same as the PSO with the difference that identification of the best particle is done by user and not by using the F.F.

Sign of trajectory of particles– In identifying particle's moving path, there are two points of view. In the positive view, which is the same as classic view, the particles adjust their positions with their best previous positions and the best global position of the swarm. In the negative view, particles adjust themselves with the worst positions, that is, they try to avoid going to the worst positions.

Recursively – In the PSO process, there are two approaches from recursively point of view. In the first view, feedback mechanism is used during the process to adapt the process with current conditions and in fact, we face with a sort of recursive PSO. But in the next view, the process is without feedback mechanism.

Hierarchy – In the hierarchy approach of the PSO, it is tried that particles be placed in a dynamic hierarchic structure in a way that particles are placed in higher levels of hierarchic structure proportional to the quality of presented solution. The higher-level particles have more effect on the total swarm.

Restriction – The PSO is divided into constrained and unconstrained types from the viewpoint of restriction. In ordinary state, which is the same as classic form of the algorithm, the algorithm is constrained. It is worth noting that in both cases the velocity and position updating equations are the same and the difference is that in the classic (constrained) case there are up and down constraints for position and velocity where if they are exceeded, those constraints will be considered. But in the UPSO case such a constraint does not exist.

Synchronicity – This algorithm has been extracted from the PSPSO algorithm. The difference between synchronicity and asynchronicity in the PSO is in position and velocity updating equations. In the PAPSO, velocity and position updating of particles is performed continuously and based on the accessible information. This algorithm makes a dynamic scheme of load balancing together with a chain duty-centered approach in order to reduce the unbalance load.

Combinatoriality – From the Combinatoriality view, the combined version of the PSO, called CPSO, is utilized to optimize combined problems including continuous and integer variables. Its opposite point is the classic PSO algorithm, which is in mere continuous state.

Cooperation – In order to improve the performance of traditional PSO, different swarms can be used cooperatively to optimize various components of the problem. This is called CPSO. Otherwise, and with a unique swarm, the uncooperative case will happen.

Objective – Considering that optimization problems are divided to single-objective and multi-objective problems from the view of objective numbers, the multi-objective and single-objective approaches have been presented to solve these kinds of problems. In the multi-objective approach, it is tried to optimize several objectives using one swarm and according to the priority of the objectives.

Compound with other heuristics – In order to increase the efficiency as well as to overcome the problems such as trapping in local optimum and in order to increase the diversity to find better solutions in the PSO, this algorithm was combined with other optimization methods such as SA, ACO, GA.

Velocity type – The velocity parameter is the main item in the PSO which specifies the direction of particle's movements. Better results can be obtained by changing this parameter using various heuristics that various examples of it are presented in the taxonomy.

Uncertainty – From the view of the information source exchanged among the swarms, in the stochastic case the information of stochastic models is employed instead of using Gbest information.

Also in [66] an overview in PSO is explained and in [88] a survey about PSO is represented. In Table III, the data related to various applications of the PSO over different years represented. Also, in Fig. II Aggregation chart for applications of the PSO over different years is introduced.

V. CONCLUSION

In this paper, the authors tried to present a general view of the PSO algorithm for researchers in this field studying the history of it and presenting various methods ramified from this algorithm as well as its various applications in different years. Since the introduction of this method in 1995, the methods branched from this algorithm and their applications have developed a lot. In this paper, based on an analysis of over 2315 publications about PSO, around 536 papers are related to methods which is improved PSO, and 1779 papers are related to PSO's applications.

In this paper, developments of the PSO are presented. Moreover, various methods derived from this algorithm are introduced so that, considering the novelty of this algorithm, they can be a guide for the researchers in the future.

Also, a table is presented in this paper expressing the main parameters used in some of the methods ramified from this algorithm so that it could be as a guide for group values to the mentioned parameters in future methodologies. Furthermore, taxonomy of this algorithm from various viewpoints was presented, which can offer a general view of this algorithm from different viewpoints. This algorithm can be a suitable tool in various optimization problems considering its more efficiency in comparison with other evolutionary algorithms such as GA and also its simplicity.

On the other hand, various applications and an analysis on these applications are evaluated. Moreover, it is tried to mention different and various applications that were introduced in the first part based on hybrid methods. In this way, just one or some applications are not discussed and it is tried to introduce the vast applications of this algorithm so that it can be useful for researchers. Thus, 41 applications of the PSO have been introduced.

The PSO algorithm, as an important algorithm in optimization, will have more applications in future in various sciences including economy, financial, business, medical science, engineering, etc. This is because of the high flexibility of the PSO.

Recently, the new methods is issued from PSO are increasingly developed. The research work on the PSO application in some fields such as electrical engineering and mathematics are widespread, but in other fields for example chemical and civil engineering are rare.

REFERENCES

[1] Aguirre, A. H., Muñoz Zavala, A. E. and Diharce, E. V., and Botello Rionda, S. (2007), "COPSO: Constraints Optimization via PSO algorithm", Comunicación Técnic, (CC/CIMAT).

[2] Alviar, J.-B., Peña, J., and Hincapié, R., (2007), “Subpopulation best rotation: a modification on PSO”, Revista Facultad de Ingeniería. No. 40, pp.118-122.

[3] Atyabi, A. and Phon-Amnuaisuk, S. (2007), " Particle swarm optimization with area extension (AEPSO)", IEEE/CEC, pp. 1970-1976.

[4] Banks, A., Vincent, J. and Anyakoha, Ch., (2007) "A review of particle swarm optimization. Part I: background and development”, Nat Comput, 6:467–484.

[5] Baskar, S. and Suganthan, P.N., (2004) "A novel concurrent particle swarm optimization,” in Proc. IEEE Cong. on Evolut. Comput., Vol. 1, pp. 792-796.

[6] Blackwell, T. M., (2003) "Swarm music: Improvised music with multi-swarms" in Proc. Symp. On Artificial Intell. and Creativity in Arts and Science, pp. 4-49.

[7] Brits, R., Engelbrecht, A. P., and van den Bergh, F. (2002), “Solving Systems of unconstrained equations using particle swarm optimization” Proc. of IEEE Conf. on Sys. Man and Cyber. (SMC), pp. 102-107.

[8] Brits, R., Engelbrecht, A.P., and van den Bergh, F., (2005), “Niche Particle Swarm Optimization”. Technical report, Department of Computer Science, University of Pretoria.

[9] Chandrasekaran, S., Ponnambalam, S.G., Suresh, R.K. and Vijayakumar, N.,(2006) “A Hybrid Discrete Particle Swarm Optimization Algorithm to Solve Flow Shop Scheduling Problems”, Proc. IEEE/ ICCIS, pp.1 – 6.

[10] Chang, J.-F., Chu, S.-C., Roddick, J.-F. and Pan, J.-S., (2005)"A parallel particle swarm optimization algorithm with communication strategies," J. of Information Science and Eng., Vol. 21, No. 4, pp. 809-818.

[11] Ching-Yi Chen, Hsuan-Ming Feng, and Fun Ye, (2007), "Hybrid Recursive Particle Swarm Optimization Learning Algorithm in the Design of Radial Basis Function Networks”, J. of Marine Science and Technology, Vol. 15, No. 1, pp. 31-40.

[12] Cui, Zh. Zeng, J. And Cai, X., (2004), “A guaranteed convergence dynamic double particle swarm optimizer”, Fifth World Cong. on Intell., Control and Automation, vol. 3, pp. 2184 – 2188.

[13] Dou, Q., Zhou, C., Pan, G., Luo, H. and Liu, Q. (2005),” Neural particle swarm optimization for casing damage prediction”, Int. Sympos. On neural networks, vol. 3498.

[14] Eberhart, R. C., and Shi, Y. (2000). "Comparing inertia weights and constriction factors in particle swarm optimization". Proc. Cong. on Evolutionary Computation, pp 84-88.

[15] Eberhart, R. C., Simpson, P. K., and Dobbins, R. W., (1996) “Computational Intelligence PC tools”, 1st ed. ed. Boston, MA: Academic press professional.

[16] El-Abd, M., (2008), "Cooperative Models of Particle Swarm Optimizers", A thesis presented to the University of Waterloo in fulllment of the thesis requirement for the degree of Doctor of Philosophy in Electrical and Computer Eng.

[17] Felix T. S. Chan1, Kumar, V. and Mishra, N., (2007), "A CMPSO algorithm based approach to solve the multi-plant supply chain problem", Swarm Intell.: Focus on Ant and Particle Swarm Optimization, pp. 532.

[18] Feng, H.-M., (2005), “Self-generation fuzzy modeling systems through hierarchical recursive-based particle swarm optimization”, J. of Cyber. and Systems, vol 36, No. 6, pp. 623-639.

[19] He, S., Wu, Q.H., Wen, J.Y., Saunders, J.R. and Paton, R.C., (2004), “A particle swarm optimizer with passive congregation”, J. of Biosystems, Vol.78, No.1-3, pp. 135-147.

[20] Higashitani, M., Ishigame, A., and Yasuda, K., (2008) “Pursuit-Escape Particle Swarm Optimization”, Trans. On Electrical and Electronic Eng., (IEEJ), Vol.3, No.1, pp.136 – 142.

[21] Ho, S.-Y., Lin, H.-S., Liauh, W.-H., and Ho, S.-J., (2008), "OPSO: Orthogonal Particle Swarm Optimization and Its Application to Task Assignment Problems" IEEE Trans. on Systems, Man and Cyber., Part A, Vol. 38, No., 2, pp.288 - 298

[22] Hu, X. and Eberhart, R. C. (2002) “Multi objective optimization using dynamic neighborhood particle swarm optimization". Proc. of the IEEE/ CEC, pp. 1677-1681.

[23] Hui, W. and Feng, Q., (2007), “Improved particle swarm optimizer with behavior of distance models”, J. of Computer Eng. and Applications, 43 (30): 30-32.

[24] Jang, W.-S., Kang, H.-I., Lee, B.-H., Kim K.-I., Shin, D.-I. and Kim, S.-C., (2007) "Optimized fuzzy clustering by predator prey particle swarm optimization" In IEEE/CEC, pp. 3232-3238.

[25] Janson, S., and Middendorf, M., (2004) "A hierarchical particle swarm optimizer for dynamic optimization problems," in Proc. Application of Evolutionary Computing, Vol. 3005, pp.513-524.

[26] Jarbouia, B., Cheikha, M., Siarryb, P. and Rebaic, A., (2007), "Combinatorial particle swarm optimization (CPSO) for partitioned clustering problem", J. Applied Mathematics and Computation, Vol. 192, Issue 2, 15 pp. 337-345.

[27] Ji, C., Zhang, Y., Gao, SH., Yuan, P. and Li, Zh., (2004), "Particle swarm optimization for mobile ad hoc networks clustering", IEEE Int. Conf. on Networking, Sensing and Control, Vol 1, pp. 372 – 375.

[28] Jie, J., Zeng, j. and Han, C., (2006), " Self-Organization Particle Swarm Optimization Based on Information Feedback", Advances in natural comput.: (Part I-II: Second Int. conf., ICNC 2006, Xi'an, China.

[29] Jingbo, A. and Hongfei, T., (2005), “Cultural based Particle Swarm Optimization”, Center for Science and Technology Development, Ministry of Education P. R. China.

[30] Jun-jie, Xu. and Zhan-hong, Xin., (2005) "An extended particle swarm optimizer" Proc. IEEE Symp., Parallel and Distributed.

[31] Kennedy, J. and Eberhart, R.C., (1995), “Particle Swarm Optimization”, Proc. IEEE Int. Conf. on N.N., pp. 1942-1948.

[32] Kennedy, J., and Eberhart, R.C. (1997), “A discrete binary version of the particle swarm algorithm”, Int. IEEE Conf. on Systems, Man, and Cyber., Vol.5, pp.4104 – 4108.

[33] Koh, B-Il, Fregly, B.-J., George, A.-D. and Haftka, R.-T., (2005), “ Parallel asynchronous particles swarm for global biomechanical”, Int J Number Methods Eng., 67(4): 578–595.

[34] Krink, T. Vesterstrom, J.S. and Riget, J., (2002), “Particle swarm optimization with spatial particle extension”, Proc. of Cong. on Evolutionary Computation, (CEC’02), Vol. 2, pp. 1474-1479.

[35] Kulkarni, R.V. and Venayagamoorthy, G.K., (2007), “An Estimation of Distribution Improved Particle Swarm Optimization Algorithm”, Proc. IEEE/ ISSNIP, pp. 539-544.

[36] Lam, H., T., Nikolaevna, P., N. and Quan N., T., M. (2007)” The Heuristic Particle Swarm Optimization” Proc. of annual Conf. on Genetic and evolutionary computation in Ant colony optimization, swarm Intell., and artificial immune systems,”GECCO’07”, pp.174 – 174.

[37] Lee, T.-Y., (2007a) “Operating schedule of battery energy storage system in a time-of-use rate industrial user with wind turbine generators: a multi pass iteration particle swarm optimization approach” IEEE Trans. on Energy Conversion, Vol. 22, No. 3, pp. 774–782.

[38] Lee, T.-Y., (2007b), “Optimal Spinning Reserve for a Wind-Thermal Power System Using EIPSO”, IEEE/ TPWRS, Vol. 22, No.4, pp. 1612 – 1621.

[39] Li, H.-Q., and Li, L., (2007), “A Novel Hybrid Particle Swarm Optimization Algorithm Combined with Harmony Search for High Dimensional Optimization Problems”, Proc. IEEE/IPC, pp. 94-97.

[40] [40] Li, T., Lai, X. and Wu, M., (2006), “An Improved Two-Swarm Based Particle Swarm Optimization Algorithm”, Proc. IEEE/ WCICA, Vol. 1, pp.3129-3133.

[41] Li, X., (2004), “Adaptively Choosing Neighborhood Bests using Species in a Particle Swarm Optimizer for Multimodal Function Optimization”, Proc. of GECCO LNCS 3102, pp.105-116.

[42] Lian, Z., Gu, X., and Jiao, B., (2006), "A Dual Similar Particle Swarm Optimization Algorithm for Job-Shop Scheduling With Penalty," in Proc. IEEE World Cong. on Intelligent Control and Automation, Vol. 2, pp. 7312-7316.

[43] Liang, J., J., Qin, A., K., and Baskar, S., (2006) ”Comprehensive Learning Particle Swarm Optimizer for Global Optimization of multimodal Functions” IEEE Trans. Evolutionary Computation, Vol. 10, No. 3.

[44] Liao, c.-y., Lee, W.-P., Chen, X., (2007), "Dynamic and adjustable particle swarm optimization, Proc. of the 8th Conf. on 8th WSEAS Int. Conf. on Evolutionary Computing, Vol. 8.

[45] Lin, C., Liu Y., and Lee, C., (2008) “An efficient neural fuzzy network based on immune particle swarm optimization for prediction and control applications”, J. of Innovative Computing, Information and Control Vol.4, No.7, pp.1711-1722.

[46] Liu, H., Ajith, A., Okkyung, C. and Hwan Moon, S., (2006), "Variable Neighborhood Particle Swarm Optimization for Multi-objective Flexible Job-Shop Scheduling Problems", Int. Conf. on Simulated evolution and learning, Seal 2006, Vol. 4247, pp. 197-204.

[47] Lu, H. and Chen, W. (2008), “Self-adaptive velocity particle swarm optimization for solving constrained optimization problems”, J. of Global Optimization, Vol.41, No.3, pp. 427-445.

[48] Lu, H., and Chen, W.,(2006), " Dynamic-objective particle swarm optimization for constrained optimization problems", J. of Combinatorial Optimization, Vol. 12, No., 4, PP. 409-419.

[49] Madar, J., Abonyi, J. and Szeifert, F., (2005), “Interactive particle swarm optimization”, Proc. Int. IEEE conf. On Intelligent Systems Design and Applications (IEEE/ ISDA), Vol.8, No.10, pp.314 – 319.

[50] McNabb, A., W., Monson, C., K. and Seppi, K., D., (2007), “MRPSO: MapReduce particle swarm optimization”, Proc. of the 9th annual Conf. on Genetic and evolutionary Comput., pp. 177 – 177.

[51] Meissner, M., Schmuker, M. and Schneider, G., (2006), “Optimized Particle Swarm Optimization (OPSO) and its application to artificial neural network training”, BMC Bioinformatics, 7:125.

[52] Mendes, R., Kennedy, J., and Neves, J., (2004) "The fully informed particle swarm: simpler, maybe better, " IEEE Trans. on Evolutionary Comput., Vol. 8 (3), pp. 204--210.

[53] Moraglio, A., Di Chio, C., Togelius, J. and Poli, R. (2008), “Geometric Particle Swarm Optimization”, J. of Artificial Evolution and Applications.

[54] Mostaghim, S., Branke, J., and Schmeck, H., (2007), "Multi-objective particle swarm optimization on computer grids" in Proc. Genetic and Evolutionary Comput. Conf., pp. 869-875.

[55] Neethling, M. and Engelbrecht, A., (2006),"Determining RNA Secondary Structure using Set-based Particle Swarm Optimization," in Proc. IEEE Cong. on Evolutionary Comput., pp. 1670-1677.

[56] Niu, B., Zhu, Y., and He, X., (2005), "Multi-population cooperative particle swarm optimization," in Proc. European Conf. on Artificial Life, M. C. et al., pp. 874-883.

[57] Noel, M.M. and Jannett, T.C., (2004), “Simulation of a new hybrid particle swarm optimization algorithm”, Proc., Of the IEEE Symp. On System Theory, pp. 150 – 153.

[58] Omkar, S. N., Mudigere, D., Narayana Naik, G., and Gopalakrishnan, S., (2008), “Vector evaluated particle swarm optimization (VEPSO) for multi-objective design optimization of composite structures”, J. Computers and Structures, Vol. 86, No.1-2. pp.1-14.

[59] Özcan, E. and Yılmaz, M., (2006), “Particle Swarms for Multimodal Optimization” Lecture Notes In Computer Science; Vol. 4431, Proc. of the 8th int. conf. on Adaptive and Natural Computing Algorithms, Part I. pp. 366 – 375.

[60] Pampara, G., Franken, N. and Engelbrecht, A.P., (2005), “Combining particle swarm optimization with angle modulation to solve binary problems,” The IEEE Cong. on Evolutionary Comput., Vol. 1, pp. 89-96.

[61] Pant, M. Radha, T., and Singh, V.P. (2007), “A New Particle Swarm Optimization with Quadratic Interpolation”, Int. IEEE Conf. on Computational Intell. and Multimedia Applications, Vol.1, pp.55-60.

[62] Pant, M., Thangaraj, R. and Abraham, A., (2008), "Particle Swarm Optimization Using Adaptive Mutation", IEEE/DEXA'08, pp. 519-523.

[63] Parsopoulos, K. E. and Vrahatis, M. N., (2004),” UPSO: A Unified Particle Swarm Optimization Scheme”, Proc. of the Int. Conf. of Computational Methods in Sci. and Eng., Vol. 1, pp. 868-873.

[64] Pasupuleti, S. and Battiti, R., (2006). “The Gregarious Particle Swarm Optimizer (GPSO)”, GECCO’06.

[65] Peram, T., Veeramachaneni, K., and Mohan, C. K. (2003), “Fitness-distance-ratio based particle swarm optimization”. Proc. of the IEEE/SIS, pp. 174-181.

[66] Poli, R., Kennedy, J., and Blackwell, T., (2007), “Particle swarm optimization an overview”, J. of Swarm Intell., 1(1), 33-57.

[67] Poli, R., Langdon, W. B., and Holland, O. (2005b). "Extending particle swarm optimization via genetic programming. In M. Keijzer et al. (Eds.), Lecture notes in computer science: Vol. 3447. Proc. of the 8th European conf. on genetic programming, pp. 291–300, Springer.

[68] Riget, J. and Vesterstroem, J. S. (2002), “A diversity-guided particle swarms optimizer - the ARPSO”. Technical Report No. 2002-02. Dept. of Computer Science, University of Aarhus, EVALife.

[69] Roy, R. and Ghoshal, S.P., (2006), “Evolutionary computation based optimization in fuzzy automatic generation control”, IEEE/ POWERI, pp.7.

[70] Sadri, J. and Suen, C.Y., (2006), “A Genetic Binary Particle Swarm Optimization Model”, Proc. IEEE/CEC, pp.656 – 663.

[71] Schoeman, I.L. and Engelbrecht, A.P., (2005), “A Parallel Vector-Based Particle Swarm Optimizer”, Proc. of the Int. Conf. on Neural Networks and Genetic Algorithms, pp. 268-271.

[72] Secrest, B.R. and Lamont, G.B. (2003), “Visualizing particle swarm optimization – Gaussian particle swarm optimization”, Proc. Of Swarm Intell. Symp. (IEEE/SIS), pp. 198- 204.

[73] Sedlaczek, K., Eberhard, P., (2006), "Using Augmented Lagrangian Particle Swarm Optimization for Constrained Problems in Engineering", J. of Structural and Multidisciplinary Optimization, Vol. 32, No. 4, pp. 277-286.

[74] Seo, J.-H., Im, C.-M., Heo, C.-G., Kim, J.-K., Jung, H.-K. and Lee, C.-G., (2006), "Multimodal function optimization based on particle swarm optimization," IEEE Trans. on Magnetics, Vol. 42, No. 4, pp. 1095-1098.

[75] Shen, X., Wei, K., Wu, D., Tong, Y. and Li, Y., (2007), “A Dynamic Adaptive Dissipative Particle Swarm Optimization with Mutation Operation”, Proc. IEEE/ ICCA, pp. 586-589.

[76] Shi, Y. and Eberhart, R., (2001). "Fuzzy Adaptive Particle Swarm Optimization", Proc. IEEE / Cong. on Evolutionary Comput..

[77] Shi, Y., and Krohling, R. A., (2002), "Co-evolutionary particle swarm optimization to solve min-max problems," in Proc. Cong. on Evolutionary Comput., Vol. 2, pp. 1682-1687.

[78] Subrarnanyam, V. Srinivasan, D. and Oniganti, R. (2007) "Dual layered PSO Algorithm for evolving an Artificial Neural Network controller", IEEE/CEC, pp. 2350-2357.

[79] Sun, J., Feng, B. and Xu, W., (2004), “Particle swarm optimization with particles having quantum behavior”, Proc. IEEE/CEC, Vol. 1, pp. 325 – 331.

[80] Gheitanchi, S., Ali, F. H. and Stipidis, E. (2008) “Trained Particle Swarm Optimization for Ad-hoc Collaborative Computing Networks” Swarm Intell. Algorithms and Applications Symp., ASIB 2008, UK.

[81] Van den Berg, F., and Engelbrecht A.P., (2004), "A cooperative approach to particle swarm optimization", IEEE Trans. on Evolutionary Comput., Vol. 8, No. 3, pp. 225-239

[82] Voss, M.S., (2005), “Principal component particle swarm optimization (PCPSO)”, Proc., Of the IEEE Symp. On swarm Intell., pp. 401 – 404.

[83] Wang, H., and Qian, F., (2007), “An improved particle swarm optimizer with shuffled sub-swarms and its application in soft-sensor of gasoline endpoint”, Proc. Advances in Intelligent Systems Research.

[84] Wang, X., Wang, Y., Zeng, H., and Zhou, H., (2006), " Particle Swarm Optimization with Escape Velocity" Int. Conf. on Computational Intell. and Security, Vol. 1, pp. 457-460

[85] Wang; X.-H. and Li, J.-J., (2004), “Hybrid particle swarm optimization with simulated annealing”, Proc., Of the IEEE Int. Conf. on Machine Learning and Cyber., Vol. 4, pp. 2402 – 2405.

[86] Wei, C., He, Z., Zhang, Y., and Pei, W. (2002). "Swarm directions embedded in fast evolutionary programming". In Proc. of the IEEE/CEC, pp. 1278–1283.

[87] Wei, K., Zhang, T., Shen, X. and Liu, J., (2007), “An Improved Threshold Selection Algorithm Based on Particle Swarm Optimization for Image Segmentation”, Proc. IEEE/ICNC.

[88] Xie, X., Zhang, W., and Yang, Z., (2002a), “Overview of particle swam optimization," Kongzhi yu Juece / Control and Decision (In Chinese).

[89] Xie, X.-F., Zhang, W.-J. and Yang, Z.-L., (2002b), “Adaptive Particle Swarm Optimization on Individual Level”, Int. Conf. On Signal Processing (ICSP), pp: 1215-1218.

[90] Xie, X.-F., Zhang, W.-J., and Yang, Z.-L., (2002c), “A Dissipative Particle Swarm Optimization”, Cong. on Evolutionary Comput. (CEC), pp.1456-1461.

[91] Xu, F., and Chen, W., (2006), "Stochastic Portfolio Selection Based on Velocity Limited Particle Swarm Optimization", Proc. IEEE/ WCICA, Vol. 1, pp.3599-3603.

[92] Xu, Xiao-ping, Fu-Cai, QIAN, and Feng W., (2008), " Research on new method of system identification based on velocity mutation Particle Swarm Optimization", (In Chinese), 44 (1) pp.31-34.

[93] Yang, C. and Simon, D., (2005), "A New Particle Swarm Optimization Technique", Proc. Of the Int. Conf. on systems Eng., (IEEE/ISEng'05).

[94] Yang, W.-P., (2007), "Vertical Particle Swarm Optimization Algorithm and its Application in Soft-Sensor Modeling", Int. Conf. on Machine Learning and Cyber. (IEEE/ ICMLC), Vol.4, pp.1985-1988.

[95] Yao, X., (2008), "Cooperatively Coevolving Particle Swarms for Large Scale Optimization", Conf. of EPSRC, Artificial Intell. Technologies New and Emerging Computer Paradigms.

[96] Yin, P. Y. (2006), "Genetic particle swarm optimization for polygonal approximation of digital curves" J. of Pattern Recognition and Image Analysis., Vol. 16, No. 2, pp. 223-233.

[97] Yuan, L. and Zhao, Z.-D., (2007), "A Modified Binary Particle Swarm Optimization Algorithm for Permutation Flow Shop Problem", IEEE/ICMLC, Vol.2, pp.902-907.

[98] Yuan, Z., Jin, R., Geng, J., Fan, Y., Lao, J., Li, J., Rui, X., Fang, Z. and Sun, J. (2005), "A perturbation particle swarm optimization for the synthesis of the radiation pattern of antenna array", Proc. IEEE Conf. on Asia-Pacific, Vol.3, No, 4-7.

[99] Moore, P.W. Venayagamoorthy, G.K., (2006), 'Empirical Study of an Unconstrained Modified Particle Swarm Optimization', IEEE/CEC, pp. 1477-1482.

[100] Zeng, J. and Hu, J. and Jie, J., (2006), "Adaptive Particle Swarm Optimization Guided by Acceleration Information", Proc. IEEE/ ICCIAS, Vol.1, pp.351-355.

[101] Zhang,Q. and Mahfouf, M., (2006) "A New Structure for Particle Swarm Optimization (nPSO) Applicable to Single Objective and Multiobjective Problems", Int. IEEE Conf. on Intelligent Systems, pp.176 – 181.

[102] Zhang, X., Hu, W., Maybank, S., Li, X. and Zhu, M., (2008), "Sequential particle swarm optimization for visual tracking", IEEE/CVPR, pp. 1-8.

[103] Zhang, Y.-N., Hu, Q.-N. and Teng, H.-F., (2008), "Active target particle swarm optimization: Research Articles", J. of Concurrency and Computation: Practice & Experience, Vol.20, No.1, pp.29 – 40.

[104] Zhao B., (2006)," An Improved Particle Swarm Optimization Algorithm for Global Numerical Optimization", Int. Conf. on Comput. Science N6, Reading, (Royaume-uni), Vol. 3994, pp. 657-664.

[105] Zhiming, L., Cheng, W. and Jian L., (2008), "Solving constrained optimization via a modified genetic particle swarm optimization", Proc. of Int. Conf. On Forensic applications and techniques in telecommunications, information, and multimedia and workshop, No. 49

First D. Sedighizadeh received the B.S. Degree in Industrial Engineering from the Amir Kabir University of Tehran, Iran and M.S. degree in Industrial Engineering from the IAU of Tehran, Iran in 2000 and 2003, respectively. Currently, he is as a PhD student in Tarbiat Modares University. His research interests Optimization, Robot Motion Planning and Intelligence Systems.

Second E. Masehian received B.S. Degree in Industrial Engineering from the Iran University of science and Technology, Tehran, Iran and M.S. degree in Industrial Engineering from the Iran University of science and Technology, Tehran, Iran and Phd degree in Industrial Engineering from the Tarbiat Modares University, Tehran, Iran in 1993, 1997 and 2004 respectively. Currently, he is an assistance professor in the faculty of engineering, Tarbiat Modares University, Tehran, Iran.

Table I. Comparison of Parameters Used in Major PSO-Based Methods

No.	Method	Author (Year)	Type	w	C_1	C_2	Remarks
1	A Parallel Vector-Based PSO (PVPSO)	Schoeman and Engelbrecht (2005)	C	0.8	1	1	
2	Active target PSO APSO	Zhang et. al. (2008)	C	0.4	2	2	
3	Adaptive Dissipative PSO (ADPSO)	Shen, et. al. (2007)	C	$0.4 \le w \le 0.9$	2	2	
4	Adaptive Mutation PSO (AMPSO)	Pant et. al. (2008)	C	–	–	–	
5	Adaptive PSO (APSO)	Xie et. al. (2002)	C	0.4	2	2	
6	Adaptive PSO Guided by acceleration information (AGPSO)	Zeng, et. al. (2006)	C	$0.1 \le w \le 0.4$	1.8	1.8	
7	Angle Modulated PSO (AMPSO)	Pampara et., al. (2005)	B	0.8	2	2	
8	Area Extension PSO (AEPSO)	Atyabi and Phon-Amnuaisuk (2007)	C	$0.2 \le w \le 1$	0.5	2.5	
9	Attractive-Repulsive PSO (ARPSO)	Riget and Vesterstrom (2002)	C	$0 \le w \le 1$	$0 \le C_1 \le 2$	$0 \le C_2 \le 2$	
10	Augmented Lagrangian PSO (ALPSO)	Sedlaczek and Eberhard (2006)	C	0.9	0.9	0.9	
11	Basic PSO (BAPSO)	Kennedy and Eberhart (1995)	C	$0.2 \le w \le 0.4$	2	2	
12	Behavior of distance PSO (BDPSO)	Hui and Feng (2007)	C	$0.4 \le w \le 0.9$	0.2	0.2	C_3, C_4 = 0.2
13	Best rotation PSO (BRPSO)	Barrera (2007)	C	0.4	2	2	
14	Binary PSO (BPSO)	Kennedy and Eberhart (1997)	D	–	–	–	
15	Chaos PSO (CPSO)	Mo et. al. (2006)	C	$0.4 \le w \le 1.2$	2	2	
16	Combinatorial PSO (CPSO)	Jarboui et. al. (2007)	C	0.95	0.6	0.4	α=7/n (n = Dimension of Search Space)
17	Comprehensive Learning PSO (CLPSO)	Liang, et. al. (2006)	C	$0.4 \le w \le 0.9$	2	2	
18	Constrained Optimization Via PSO (COPSO)	Aguirre, et. al. (2007)	C	$0.5 \le w \le 1$	1	1	
19	Cooperative Co evolutionary PSO CCPSO	Yao (2008)	C	–	–	–	
20	Cooperative Multiple PSO (CMPSO)	Chen, et. al. (2007)	D	–	–	–	

No.	Method	Author (Year)	Type	w	C_1	C_2	Remarks
21	Cultural Based PSO (CBPSO)	Jingbo and Hongfei (2005)	C	–	–	–	
22	Discrete PSO (DPSO)	Kennedy and Eberhart (1997)	D	–	1.2	1.2	$v_{\max} = 4$
23	Dissipative PSO (DPSO)	Xie et. al. (2002)	D	$0.4 \leq w \leq 0.9$	2	2	
24	Divided Range PSO (DRPSO)	Ji, et. al. (2004)	D	$0.1 \leq w \leq 1.2$	2	2	
25	Double-structure coding Binary PSO (DS-BPSO)	He, et. al. (2007)	D	–	–	–	
26	Dual Layered PSO (DLPSO)	Subrarnanyam, et. al. (2007)	C	–	–	–	$C_1 = C_1 + (n / Iter_{max})$ $C_2 = C_2 - (n / Iter_{max})$
27	Dynamic & Adjustable PSO (DAPSO)	Liao, et. al. (2007)	C	–	–	–	
28	Dynamic Double PSO (DDPSO)	Cui, et. al. (2004)	C	$0.4 \leq w \leq 1$	1.8	1.8	
29	Dynamic Neighborhood PSO (DNPSO)	Hu et. al. (2003)	C	$0.5+(rand/2)$	1.4944	1.4944	
30	Escape Velocity PSO (EVPSO)	Wang, et. al. (2006)	C	0.7	1.49	1.49	
31	Estimation of Distribution PSO (EDPSO)	Kulkarni, et. al. (2007)	C	$0.4 \leq w \leq 0.9$	2	2	
32	Evolutionary Iteration PSO (EIPSO)	Lee (2007)	C	50	30	44	$C_2 = C_1\ (1-e^{-C_1 \cdot k})$
33	Evolutionary Programming PSO (EPPSO)	Ye, et. al. (2007)	C	$0.1 \leq w \leq 0.35$	1.5	1.5	
34	Evolutionary PSO (EPSO)	Miranda and Fonseca (2002)	C	$w^* = w + \tau .\ N(0, 1)$ $0.4 \leq w \leq 0.9$	–	–	τ is parameter Learning and N (0, 1) is a random variable with Gaussian Distribution
35	Exploring Extended PSO (XPSO)	Poli et. al. (2005)	C	0.7	1	1	
36	Extended PSO (EPSO)	Jun-jie and Zhan-hong (2005)	C	0.729	–	–	$\varphi_1 + \varphi_2 + \varphi_3 = 4.1$
37	Fast PSO (FPSO)	Li et. al. (2007)	C	0.729844	1.49618	1.496180	C_1,C_2 are η_1, η_2
38	Fully informed PSO (FIPS)	Mendes, R.; Kennedy, J. and Neves, J. (2004)	C	0.7298	2.05	2.05	
39	Fuzzy PSO (FPSO)	Shi and Eberhart (2001)	C	3, 0.2, 1.1	2	2	
40	Gaussian PSO (GPSO)	Secrest and Lamont (2003)	C	–	0.4	0.6	
41	Genetic Binary PSO (GBPSO)	Sadri and Suen (2006)	B	–	–	–	α =0.0135, β=0, θ=25π/16
42	Genetic PSO (GPSO)	Yin (2006)	C	$0.4 \leq w_1 \leq 0.9$, $w_2 = 0.2 \times w_1 + 0.8$,	–	–	P_m = 0.001 (mutation prob.)
43	Geometric PSO (GPSO)	Moraglio et. al. (2008)	C	$0.3 \leq w \leq 0.7$	–	–	
44	Greedy PSO (GPSO)	He et. al. (2007)	C	–	–	–	
45	Gregarious PSO (GPSO)	Srinivas Pasupuleti, Roberto Battiti (2006)	C	–	–	–	$\gamma_{init.} = 3.0$, $-2.4 \leq \delta \leq 0.5$
46	Heuristic PSO (HPSO)	Lam et. al. (2007)	C	0	–	–	$0.2 \leq \varphi_1, \varphi_2 \leq 2.25$
47	Hierarchical Recursive-based PSO (HRPSO)	Feng (2005)	C	0.75	1.5	1.5	
48	Hybrid Discrete PSO (HDPSO)	Chandrasekaran, et. al. (2006)	B	0.2	2	2	
49	Hybrid Gradient PSO (HGPSO)	Noel and Jannett (2004)	C	–	–	–	
50	Hybrid Recursive PSO HRPSO	Chen et. al. (2007)	C	0.75	1.5	1.5	
51	Hybrid Taguchi PSO (HTPSO)	Roy and Ghoshal (2006)	C	0.2	2.05	2.05	
52	Immune PSO (IPSO)	Lin et. al. (2008)	C	$0.4 \leq w \leq 0.9$	2	2	
53	Improved PSO (IPSO)	Zhao (2006)	C	$0.4 \leq w \leq 0.9$	0.8	0.8	C_3 = 0.6 (Congregation coef.)
54	Interactive PSO (IPSO)	Madar et. al. (2005)	C	$w=0.4+((Iter_{max}-i)/(2*Iter_{max}))$	L. F.	L. F.	L.F. = Learning Factor
55	Map-Reduce PSO (MRPSO)	McNabb, et. al. (2007)	C	$0.2 \leq w \leq 0.4$	2	2	
56	Modified Binary PSO MBPSO	Yuan and Zhao (2007)	B	0.33	0.33	0.33	
57	Modified GPSO (MGPSO)	Zhiming et. al. (2008)	C	0.4	2	2	
58	Nbest PSO	*Brits* et. al. (2002)	C	$0.1 \leq w \leq 0.5$	2	2	Decreasing Linearly w
59	Neural PSO (NPSO)	Dou, et. al. (2005)	C	0.5	1.4	1.4	
60	New PSO (NPSO)	Yang et. al. (2005)	C	$0.2 \leq w \leq 0.4$	2	2	C_1, C_2 are similar to BAPSO
61	New PSO (NPSO)	Zhang, Q. and Mahfouf, M. (2006)	C	–	–	–	
62	Niche PSO	*Brits* et. al. (2002)	C	$0.2 \leq w \leq 0.7$	1.2	1.2	Decreasing w
63	Novel Hybrid PSO (NHPSO)	Li and Li (2007)	C	0.729	1.4944	1.4944	
64	Novel PSO (NPSO)	Zhu, et. al. (2008)	D	–	–	–	

No.	Method	Author (Year)	Type	w	C_1	C_2	Remarks
65	Optimized PSO (OPSO)	Meissner et. al. (2006)	C	$w=w_{Start}-((w_{Start}-w_{End})/Iter_{max})\times i$	$C_2/C_1=2.14$		C_2/C_1 value is for Rosenbrock function. $w_{start} < 0$, $w_{end} > 0$
66	Orthogonal PSO (OPSO)	Ho et. al. (2008)	C	0.9	2	2	
67	Parallel Asynchronous PSO (PAPSO)	Koh, B., et. al. (2006)	C	Dynamic	2	2	
68	Perturbation PSO (PPSO)	Yuan et. al. (2005)	C	$0.1 \le w \le 0.9$	1.5	1.5	–
69	Predator Prey PSO (PPPSO)	Kim (2007)	C	$w= (Iter_{Total} - Iter_{Curr.})/Iter_{Total}$	2	2	
70	Principal Component PSO (PCPSO)	Voss (2005)	C	$0.4 \le w \le 0.7$	2	2	
71	PSO with Craziness and Hill Climbing (CPSO)	Ender Özcan, Murat Yılmaz (2006)	C	0.729	2.05	2.05	
72	PSO with Passive Congregation (PSOPC)	He et. al. (2004)	C	$0.7 \le w \le 0.9$	0.5	0.5	
73	Pursuit-Escape PSO PEPSO	Higashitani (2008)	C	0.729	1.4955	1.4955	
74	Quadratic Interpolation PSO (QIPSO)	Pant et. al. (2007)	C	0.9	2	2	w decreases linearly
75	Quantum Delta PSO (QDPSO)	Sun, et. al. (2004)	C	–	–	–	$g = 0.96$
76	Quantum PSO (QPSO)	Yang et. al. (2004)	C	–	–	–	
77	Quantum-Inspired PSO (QIPSO)	Sun., et. al. (2004)	C	–	0.1	0.1	$C_3 = 0.8$, $\alpha = 0.1$, $\beta = 0.9$
78	Restricted Velocity PSO (RVPSO)	Lu and Chen (2006)	C	1	1	1	
79	Self-adaptive velocity PSO (SAVPSO)	Lu and Chen (2008)	C	1/2	1	1	
80	Self-Organization PSO (SOPSO)	Jie, et. al. (2006)	C	0.7298	1.4962	1.4962	
81	Simulated Annealing PSO (SAPSO)	Wang and Li (2004)	C	The parameter w reduces gradually as the generation increasing	2	2	
82	Spatial Extension PSO (SEPSO)	Krink, et. al. (2002)	C	$0.6 \le w \le 0.9$	2	2	
83	Special Extension PSO (SEPSO)	Monson and Seppi (2006)	C				
84	Species Based PSO (SPSO)	Li et. al. (2004)	C	–	–	–	
85	Sub-Swarms PSO(SSPSO)	Wang and Qian (2007)	C	$0.4 \le w \le 0.9$	$C_1 + C_2 = 4$		
86	Trained PSO (TPSO)	Gheitanchi et. al. (2007)	C	–	–	–	
87	Two-dimensional Otsu PSO (TOPSO)	Wei, et. al. (2007)	C	0.729	2.05	2.05	
88	Two-Swarm PSO (TSPSO)	Li, et. al. (2006)	C	0.729	2.05	2.05	
89	Unconstrained PSO (UPSO)	Moore and Venayagamoorthy (2006)	C	0.8	2	2	
90	Unified PSO (UPSO)	Parsopoulos and Vrahatis (2004)	C	–	2.83	2.83	$\chi = 0.6$
91	Variable Neighborhood PSO (VNPSO)	Wang, et. al. (2006)	C	$0.1 \le w \le 0.9$	1.49	1.49	
92	Vector Limited PSO (VEPSO)	Omkar, et. al. (2008)	C	0.9	1.494	1.494	
93	Velocity Limited PSO (VLPSO)	Xu and Chen (2006)	C	0.9	1.494	1.494	
94	Velocity Mutation PSO (VMPSO)	Xu, et. al. (2008)	C	0.9	1.494	1.494	
95	Vertical PSO (VPSO)	Yang (2007)	C	$0.2 \le w \le 0.4$	2	2	

Legend: B = Binary; C = Continuous; D = Discrete.

Table II. Major PSO Bibliographies

UPDATE DATE	DEVELOPER	NUMBER OF CITED WORKS	WEBPAGE ADDRESS
1998	ENDER ÖZCAN	15	HTTP://CSE.YEDITEPE.EDU.TR/~EOZCAN/RESEARCH/PSO/PSO_BIB.HTML
2001	A. CARLISLE AND G. DOZIER	39	HTTP://WWW.ENGR.IUPUI.EDU/~SHI/PSO/BIBLIOGRAPHY.HTML
2002	M. A. ABIDO	106	HTTP://WWW.COMPUTELLIGENCE.ORG,
2004	X. HU AND R. EBERHART	316	HTTP://WWW.SWARMINTELLIGENCE.ORG/BIBLIOGRAPHY
2005	UNIV. CARLOS III, MADRID	27	HTTP://WWW.UC3M.ES/PORTAL/PAGE/PORTAL/INICIO
2005	JACO F. SCHUTTE	41	http://www.tech.plym.ac.uk/spmc/links/pso/pso_papers.html
2006	ANDRIES P. ENGELBRECHT	255	HTTP://WWW.SIMONGARNIER.COM/SWARM_REFERENCES/?PAGE=PUBLICATION&KIND=LIST&TYPE=YEAR

Table III. The data related to various applications of the PSO over different years

Application	Year														Total
	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	
Electrical Engineering															
Electricity generation and power systems	-	-	-	-	1	3	6	11	9	11	25	64	45	36	211
Design and control of neural networks	-	-	1	-	-	3	2	4	2	11	14	27	32	32	128
Control applications	-	-	-	-	-	-	1	3	3	8	18	41	24	30	128
Design and control of fuzzy systems	-	-	-	-	-	-	1	1	3	7	14	18	17	24	85
Electronics and electromagnetic	-	-	-	-	-	-	-	1	6	15	13	15	12	14	76
Design and optimization of communication networks	-	-	-	-	-	-	-	-	-	9	16	21	11	19	76
Image and sound analysis	-	-	-	-	-	-	-	-	2	5	3	10	21	11	52
Antenna design	-	-	-	-	-	-	-	-	2	2	12	12	12	11	51
Design and restructure of electricity networks and economic load dispatching	-	-	-	-	-	-	1	1	4	4	3	16	12	-	41
Sensor networks	-	-	-	-	-	-	-	1	1	2	5	6	12	13	40
Design and optimization of electric motors	-	-	-	-	-	-	-	-	3	1	-	9	6	4	23
Design and control of fuzzy-neural networks	-	-	-	1	-	-	-	-	1	1	2	3	6	6	20
Filter design	-	-	-	-	-	-	-	-	-	-	-	9	4	4	17
Unit commitment		-	-	-	-	-	-	-	1	-	3	6	3	2	15
Fault detection and recovery	-	-	-	-	-	-	-	-	1	1	-	8	1	2	13
Computer Science and Engineering															
Visualization and computer graphics	-	-	-	-	-	-	-	-	2	-	6	4	1	7	20
Making music and games	-	-	-	-	-	-	-	-	1	2	3	3	1	1	11
Mechanical Engineering															
Robotics	-	-	-	-	-	-	-	1	1	5	9	22	19	17	74
Dynamic systems	-	-	-	-	-	-	1	-	2	4	2	4	1	4	18
Industrial Engineering															
Scheduling	-	-	-	-	-	-	-	-	3	4	3	22	19	25	76
Sequencing	-	-	-	-	-	-	-	-	-	1	12	-	2	3	18
Forecasting	-	-	-	-	-	-	-	-	1	2	3	8	11	8	33
Maintenance planning	-	-	-	-	-	-	-	-	-	-	-	1	2	5	8
Job and resource allocation	-	-	-	-	-	-	-	-	-	-	-	-	3	4	7
Supply chain management	-	-	-	-	-	-	-	-	-	-	-	1	1	3	5
Civil Engineering															
Civil engineering	-	-	-	-	-	-	-	-	-	-	-	4	-	1	5
Traffic management	-	-	-	-	-	-	-	-	1	-	-	1	1	2	5
Chemical Engineering															
Chemical process	-	-	-	-	-	-	-	-	1	-	4	4	2	4	15
Mathematics															

Data mining	-	-	-	-	-	1	-	2	4	14	18	33	35	48	155
Multi objective optimization	-	-	-	-	-	-	-	2	4	4	16	16	30	25	97
Optimization of constrained problems	-	-	-	-	-	-	-	2	1	4	2	7	16	6	38
Multi model function	-	-	-	-	-	-	-	-	3	2	1	3	6	4	19
Modeling	-	-	-	-	-	-	-	-	-	4	4	1	5	5	19
Traveling salesman problem	-	-	-	-	-	-	-	-	1	3	4	-	-	2	10
Combinational optimization	-	-	-	-	-	-	-	-	-	-	1	-	1	2	4
Other Applications															
Miscellaneous	-	-	-	1	1	-	-	1	4	5	5	11	15	11	54
Economical and financial applications	-	-	-	-	-	-	1	2	-	5	8	13	3	11	43
Biological and medical applications	-	-	-	-	1	-	-	-	1	5	1	10	2	8	28
System identification	-	-	-	-	-	-	-	-	-	2	1	9	8	6	26
Materials engineering	-	-	-	-	-	-	-	-	1	3	3	1	2	2	12
Security and military applications	-	-	-	-	-	-	-	-	-	-	-	2	-	1	3
Total	0	0	1	2	3	7	13	32	69	146	234	445	404	423	1779

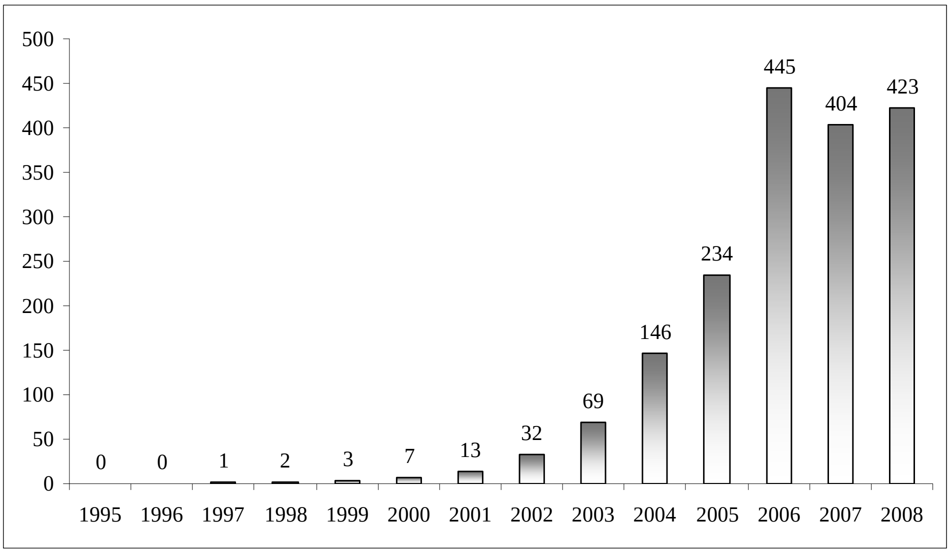

Fig II. Aggregation chart for applications of the PSO over different years

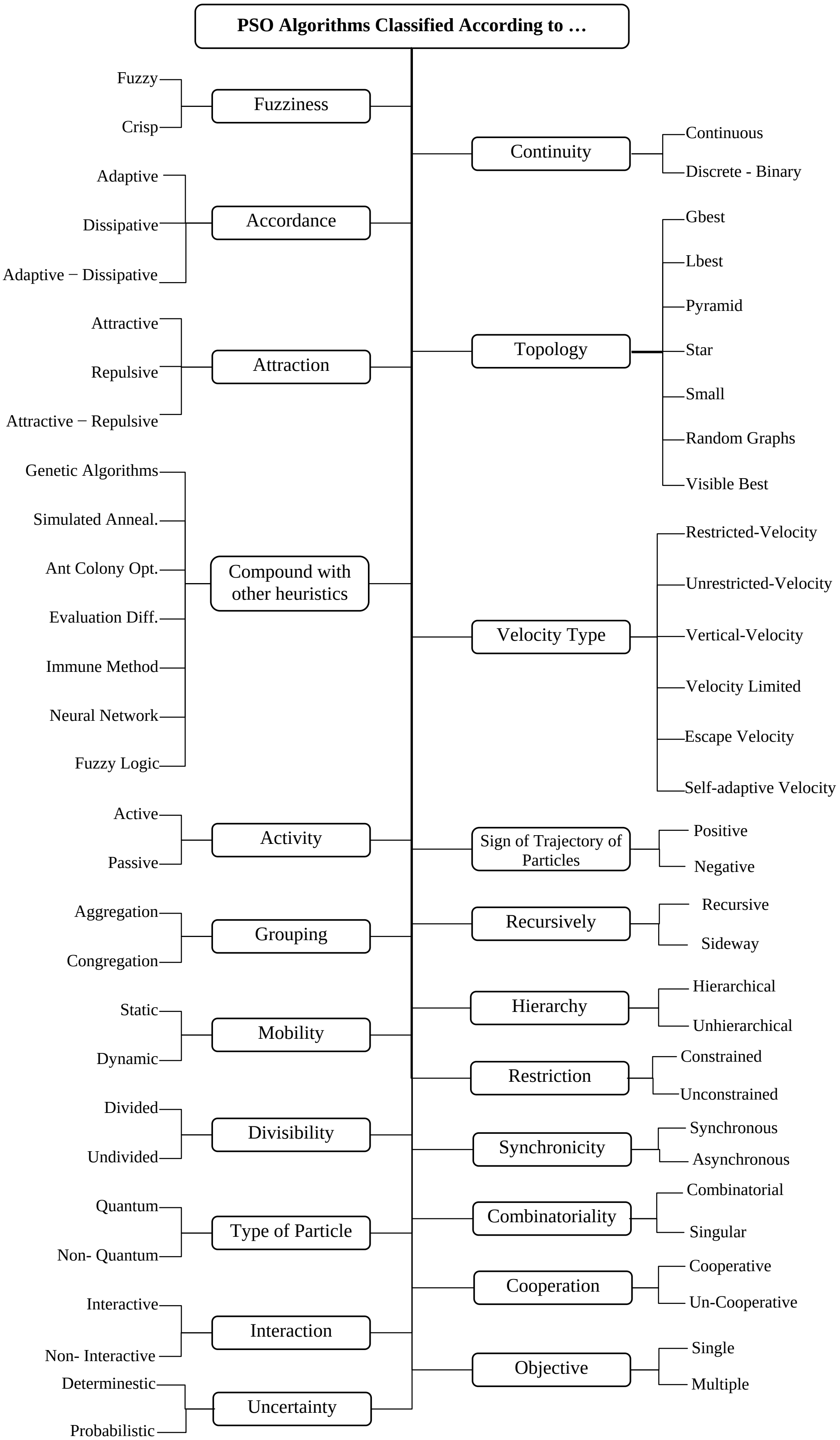

Fig. I. Taxonomy of PSO