Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 6, July 2021 : 5879 - 5895

Research Article

A Comparative Analysis on Assorted Versions of Particle Swarm Optimization Algorithms: BPSO, DPSO, PSO-DE, PSO-NE and HPSO

Amir Sohel, [a*], Utpal Chandra Das [b]

[a*] Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh
[b] Electrical Engineering, Indian Institute Of Technology Roorkee, India

[a*]**Corresponding author:** uc_das@ee.iitr.ac.in

Abstract

Particle Swarm Optimization (PSO) is a meta-heuristic algorithm to determine the social behavior of different insects and to implement their optimization techniques in various real-life optimization problems. Swarms use these optimization techniques to lead their daily life for communicating with other swarms, foraging and mating activities and travelling to their destination. PSO algorithms used these methods to solve optimization problems in several fields of computer science. However, real-life problems have different parameters and problem space is not the same for all. That's why the original version of PSO cannot be applied in every problem. PSO algorithm changed its needful features to solve new problems based on problem specification. Different versions of PSO come up with different optimization methods focusing on specific problem space. In each version, many methods or approaches have been proposed by many researchers and their optimization strategy and performance of finding an optimal solution can be different. A study on extensive analysis of different variants of PSO algorithms should have a great value for prospective researchers focusing in this field. In this research, we will explain optimization techniques of assorted versions of PSO algorithm (BPSO, DPSO, PSO-DE, PSO-NE and HPSO) and their variants. Then analyze their performance and present comparative results. We have compared the performance of each variant in terms of benchmark functions and other related algorithms. From this research, we will have acquainted which version is better for which type of optimization problems..

Keywords:

1. Introduction

From the beginning of life, swarms used their different social behaviours to lead their life. Swarms have good communication between each other by different individual behaviours such as phenomenon, waggle dance, frequency and so on. Swarm Intelligence plans and implements different effective computational techniques for solving complex problems in a way that is enlivened by the conduct of swarms or swarm's colonies. SI has been used for both combinatorial and continuous optimization problems in both dynamic and static search space [1]-[4].

Particle swarm optimization (PSO) is a branch of Swarm Intelligence (SI) that works for optimization problems simulated from the social behaviours of different swarms. PSO algorithm is very effective in solving problems with nonlinearity and non-differentiability features, multiple optima and high dimensionality. PSO algorithm has been successfully applied in both discrete and continuous optimization problems and showed promising performance compared to other optimization algorithms. Nowadays PSO uses groups of swarms rather than single swarm so that different parts of search space are discovered by different swarms to improve exploration and exploitation abilities.

There are many variants of particle swarm optimization algorithms which are based on particles behaviours, characteristics, optimization methods and characteristics of problem space. For specific problem space, PSO provides a particular version of optimization technique and each version has many methods of optimization for different specific problems of that version. Performance of each version is divergent from others in respect of detecting optimal solution and computation time. So analysis between them is necessary to find the appropriate algorithm for solving upcoming problems in optimization.

Figure 1: Honey Bee Swarm

Figure 2: Ant Swarm

In this research we explain five variants of PSO algorithm named as Binary PSO (BPSO), Discrete PSO (DPSO), PSO for Dynamic Environment (PSO-DE), PSO for Noisy Environment (PSO-NE), and Hybrid PSO (HPSO). BPSO used real number spaces for optimization. DPSO used for discrete valued problem space with finite variables. PSO-DE applied in a continuously changing environment. PSO-NE applied to real life problems which included noise. And HPSO was made by the combination of different PSO algorithms for solving new problems. We also explain their optimization technique and performance.

The rest of the paper is arranged as follows. In Section 2, details of Particle Swarm Optimization. Section 3 describes the Binary Particle Swarm Optimization. Discrete Particle Swarm Optimization described in Section 4. Particle Swarm Optimization for Dynamic Environments and Noisy Environment will describe in Sections 5 and 6 respectively. Section 7 describes the Hybrid Particle Swarm Optimization. Discussion of results in Section 8 and the conclusion in Section 9.

2. Particle Swarm Optimization

Particle swarm optimization (PSO) is a population based metaheuristic algorithm proposed by Kennedy and Eberhart (1995) based on social behavior of bird's flock [5]. Later researchers found many optimization algorithms based on simulation of social behaviors of other swarms such as honey bee, mosquito, cockroach, bat, bacteria and many more; which are able to solve different kinds of real life problems. The PSO is made up with large number of simple interacting particles. Particles have memory and also have the ability to decide when they need to be updated. PSO randomly initializes its initial population position and velocity [6].

The behavior of PSO depends on the structure of their fitness function. In PSO *d*-dimensional search space, particles change their position based on their current position and the global best (best position found so far). Particles compare their position with the new best position at every iteration. If the fitness value of the new position is better than the current position then, replace the new fitness value with the current position. There is a g-best topology (global best) which is an example of static topology where the target particle is influenced by the best neighbor and that may act like a connected graph which keeps record of the best function result that has been found. Another static topology is L-best (local best) which can perform parallel search and converge to a similar region of the search space so that the population can find the local optima from equally good optima. Here, particles have information of their own and their best neighbor. Particles move towards the best neighbor based on the local best (l-best) and the global best (g-best) instead of current best and global best. If there are two neighbors then, compare the fitness value and choose the best one.

In PSO, inertia weight is implemented for exploration and exploitation control and also used to adapt fuzzy systems. It is also used to improve the performance in applications. Large inertia weight simplifies the global exploration where small inertia weight simplifies the local exploration. So we need a balanced inertia weight for balance among local and global exploration attributes and reduce the number of iterations for locating optimal solution [7][8].

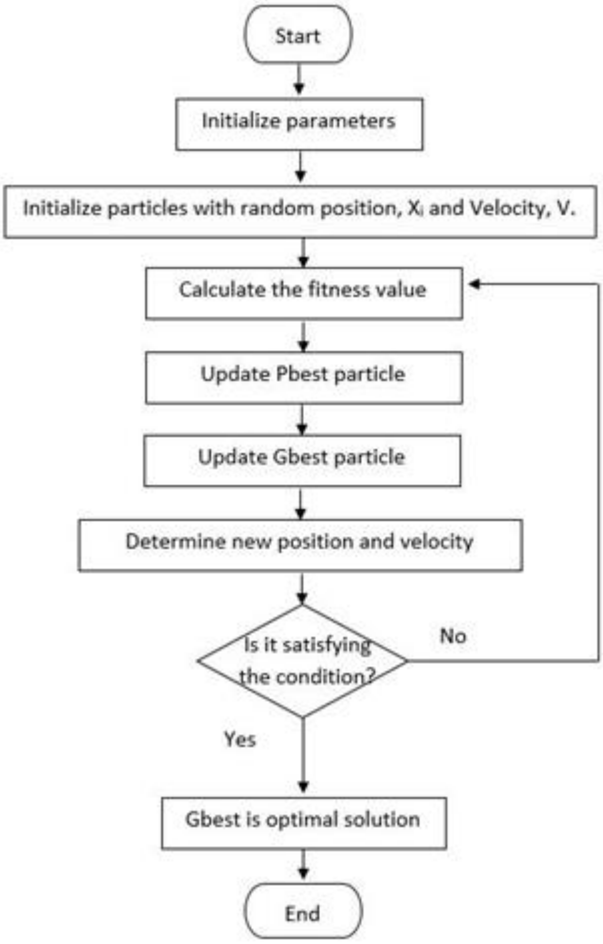

Figure 3: Flow chart of fundamental PSO algorithm

3. Binary Particle Swarm Optimization (BPSO) Algorithms

In this section at first we discuss the Elementary BPSO algorithm with flowchart. Then we present few important variants of BPSO algorithm and its applications followed by their comparative analysis.

3.1. Elementary BPSO Algorithm

Binary particle swarm optimization algorithm was one of the first algorithm that provided global optima of every experimentation of problem and also faster than GA (genetic algorithm) in both crossover and mutation problems except problems with low dimension and less number of local optima. In binary PSO, each particle is represented by zero or one. The velocity of the particle in binary PSO depends on the probability of the particle that might change its state from zero to one and vice versa. In discrete binary space, update of position means to switch the value from 0 to 1 or 1 to 0 based on the velocity of agents. Position updates in binary space depend on the probability of its velocity. For position updating, transfer function is needed to map the probability values from velocity and the probability of position vector. There are two main problems of binary PSO. I) Parameter of the binary PSO is different for real valued problems. 2) Memory of the binary PSO can store the current value and update its value by comparing current best and new best. In continuous PSO, large numbers of velocity encourage for exploration but in binary PSO small numbers of velocity encourage for exploration.

Algorithm 1: Elementary BPSO algorithm:

Step 1: Initialize the Swarm X_i (Random value between 0 and 1)

Step 2: Evaluate fitness value, f for each particle based on X_i

Step 3: Compare the fitness value of each particle to its personal best.

If $f(X_i(t)) < f(V_{lbest})$

$f(V_{lbest})=f(X_i(t))$

$V_{lbest} = X_i(t)$

Step 4: Compare the fitness value of each particle with global best.

If $f(X_i(t)<f(V_{gbest})$

$f(V_{gbest})=f(X_i(t)$

$V_{gbest} = X_i(t)$

Step 5: Change the velocity of the particle in 0 and 1.

Step 6: Calculate the changes of the bits of velocity.

Step 7: Generate a random variable R in the range (0,1) and move each particle to a new position.

Step 8: Go to step 2, and repeat until convergence.

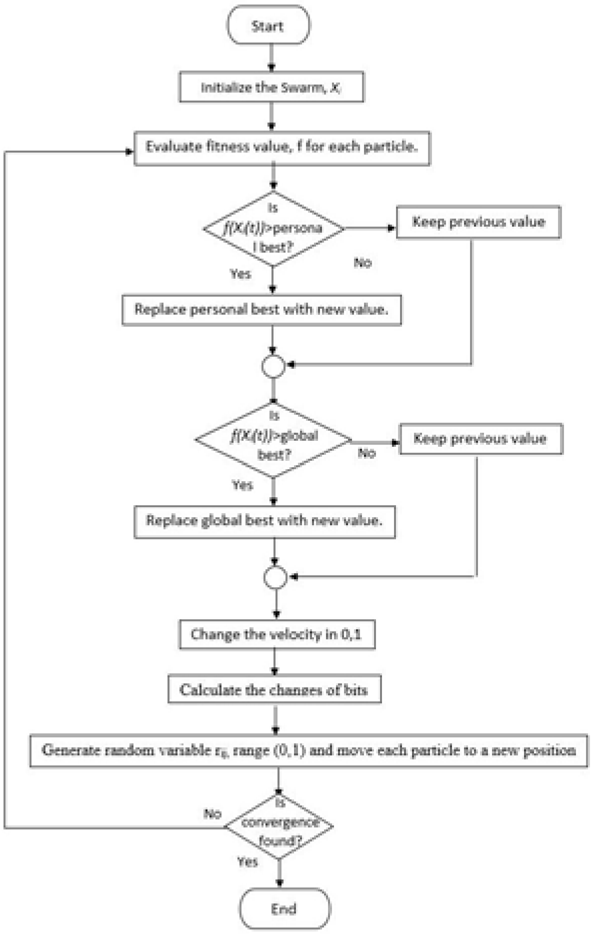

Figure 4: Flow chart of Binary PSO algorithm.

3.2. Variants and Applications

Kennedy and Eberhart [1997] proposed a binary PSO algorithm (DBPSO) for discrete space to adjust the paths of particles to address problems including floating-point, ordering or arranging of discrete elements in scheduling, and routing problems. Binary space can be considered as a hypercube where particles are moved in search space by flipping a various number of bits such as velocity, the number of bits changed per iteration or hamming distance. A particle cannot move without flipping bits, whereas it needs to reverse all of its binary coordinates for farthest movement [8]. M. F. Taşgetiren and Yun-Chia Liang [2003] also proposed A binary version of the PSO algorithm combined with the global version of PSO for the lot-sizing problem [10].

M. A. Khanesar, M. Teshnehlab [2007] proposed a novel binary PSO (NBPSO) to interpret continuous PSO into discrete PSO and introduced velocity vector in binary PSO. In this method, each particle swarm is represented by a vector in multidimensional search space where a vector has vector velocity which determines the next movement of the particle. Here, the PSO algorithm updates the velocity vector through updating the velocity of each particle based on the current position, global best position and best position explored till now [11].

S. Mirjalili, S.Z. Mohan Hashim [2011] proposed a binary version of PSO for various transfer functions BPSO-TF). This method can avoid trapping into local optima and overcome the slow concurrence rate of PSO algorithm. They used the V-shape family of transfer function instead of the S-shape family to enhance the accuracy of the result. For binary PSO, transfer function in the interval of [0,1] and increase with the rise of velocity. For large velocity, transfer function provides a high probability of changing position whereas it provides small probability for small velocity [12].

SL. Gupta, Anurag Sing [2019] proposed a binary feature selection method based on scale-free topological structures which have the capabilities to tackle the high-dimensional datasets with better classification accuracy at the lowest number of features. The Scale-free topological network follows the principle of growth and preferential attachment. Scale-free binary PSO based on feature selection method (SF-BPSO) outperforms conventional binary PSO algorithm in selecting relevant features and maintaining the high level of classification accuracy through testing on six high-dimensional datasets or validation of the model. SF-BPSO successfully reduced features with higher classification accuracy compared to conventional BPSO for a large number of categories [13].

Hongwei Cai, Xue Li, Chungang Xie [2019] proposed a method that used binary PSO for proper distribution of conductive materials in problem space that helps to reduce the heat of different electronic devices. The Binary version can easily represent the distribution of conductive material by 0 or 1 which means either exists or not. Here, 1 represents highly conductive material and 0 for heat-generating elements. Conducting path can be identified by the value of the phenotype [12].

In table 1 we present a summarized comparison among major variants of BPSO algorithm.

Algorithms	**Comparison with Benchmark Functions**	**Comparison with Other Algorithms**	**Performance**
DBPSO	Not compared	Outperforms original BPSO and GA	Extremely Flexible and Robust.
NBPSO	Satisfy Sphere, Rosenbrock, Griewangk and Rastrigin benchmark functions.	Outperforms DBPSO of kennedy and BPSO proposed by M. faith Tasgetiren & Yun-chia liang	Removing inertia weight makes the algorithm faster and leads to a better solution.
BPSO-TF	Satisfy Spherical, Rastrigin, Rosenbrock, and Griewangk benchmark function.	Outperforms original PSO and BPSO algorithm.	Overcome the problem of being trapped into local optima and slow convergence rate problem of PSO algorithm.
SF-BPSO	Not compared	Outperforms original BPSO for high-dimensional datasets.	Provide better classification accuracy on feature selection of high dimensional datasets.

Table 1: Comparison among major BPSO variants

4. Discrete Particle Swarm Optimization (DPSO) Algorithm

Elementary BPSO algorithm and flowchart will discuss in this section. After that we present few important variants of BPSO algorithm and its applications followed by their comparative analysis.

4.1. Elementary DPSO Algorithm

The PSO was mainly designed for continuous-valued space but for some problems it is defined as discrete-valued space in which variables are finite. In the discrete version, it comes up with a probability threshold P_t. If P_t remains fixed, the position of the particle is still dynamic on that dimension because it flips polarity with the probability of P_t. Discrete PSO (D-PSO) algorithms are black-box optimization algorithms. For this, they don't require any problem-specific knowledge. Runtime results of discrete PSO algorithms are available for the pseudo-Boolean function which contains all functions from the set of bit-strings to the real number. Single-source shortest path problem which calculates the length of the shortest path from source to destination and permutation problems that ask for the minimal or maximal value of a function. In discrete PSO algorithm, the fitness level method is a versatile technique to reduce runtime for randomized search problems. For pseudo-Boolean function, D-PSO algorithm reaches the optimal solution with a constant number of particles. Better runtime obtained by Analysis of D-PSO for permutation problems by characteristics of objective function and neighbourhood search technique which is given by transpositions of particles. In the algorithm, rand() means random value.

Algorithm 2: Elementary DPSO Algorithm:

Step 1: Initialize population *P* and discrete particles *D*.

Step 2: Calculate the fitness of discrete particles *D*.

Step 3: Calculate $P_{localbest}$

Step 4: Calculate $P_{globalbest}$

Step 5: Compute probability of *P*.

Step 6: Calculate discrete particles *D*, If $rand > P_l^j$, then $D_l^j = 1$; else $D_l^j = 0$.

Step 7: Back to step 2, until one of the stopping criteria is satisfied.

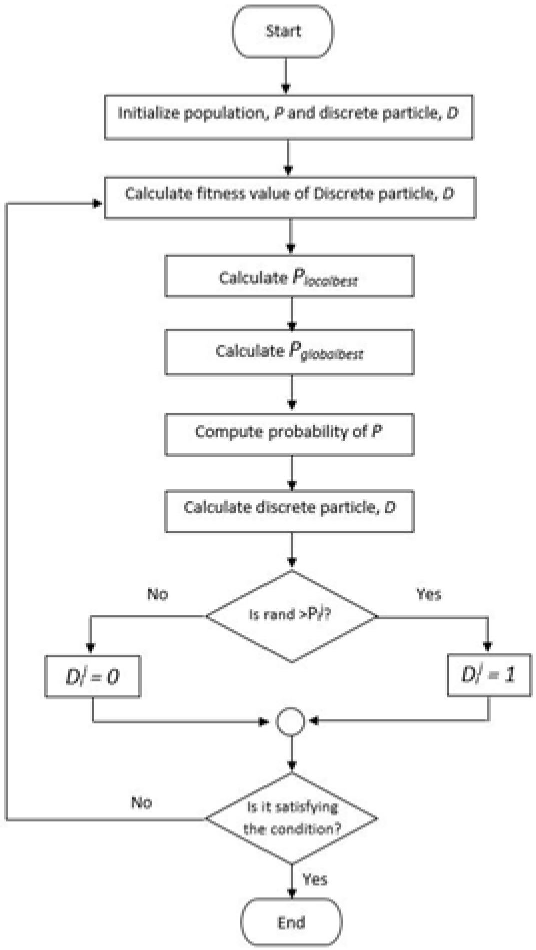

Figure 5: Flow chart of Discrete PSO algorithm

4.2. Variants and Applications

Zwe-Lee Gaing [2003] proposed a discrete PSO algorithm for unit commitment. Unit commitment is used to determine the optimal set of units that are present in the scheduling period and how long they exist. The motive of the unit commitment is to reduce total generation cost which includes production cost and transition cost. BPSO algorithm and lambda-iteration methods are used to solve unit scheduling and economic dispatch problems respectively and satisfy the goal of unit commitment. The best particle of BPSO has the lowest total generation cost [15].

Wei Pang, Kang-ping Wang [2004] proposed a fuzzy discrete PSO algorithm for solving traveling salesman problem (TSP) which is NP-Hard complete problem to find the minimal length of route for visiting the whole city once. The velocity and position of PSO represented as a fuzzy matrix and then normalized the position matrix. When the position matrix shows the potential solution of TSP then we need to decode and get a feasible solution by de-fuzzification and find the route solution value from the route array and the cost of the length of TSP is the value of position matrix [16]. They also proposed a modified PSO method in transforming the environment such as continuous space to discrete permutation space to solving TSP problems with the help of local search and chaotic operations [17].

B. Al kazemi, C.K. Mohan [2005] proposed a discrete multi-phase PSO algorithm (DiMuPSO) that utilizes the groups of particles in a different phase and increases the diversity with different goals which change over time. Particles move towards the recent best solution so that it can improve fitness. Sometimes changing in direction of search leads to the goal so that every particle changes their directions if they found no improvements in their fitness for a certain period. It used a hill-climbing algorithm to avoid being trapped into local optima and obtaining global optima [18].

CHEN Ai-ling, YANG Gen-ke, WU Zhi-ming [2005] introduce a hybrid discrete PSO (DPSO-SA) for capacitated routing problem (CVRP) which is NP-Hard problem that determines the routes simultaneously for several vehicles from central repulse to customers and again back to repulse without accessing the fixed capacity of each vehicle. We have to minimize the travelling distance of vehicles to reduce the cost by application of DPSO in large scale problems. The Discrete point has a value of 0 or 1. If the value is 1 that means the corresponding customer is served by the relevant vehicle. Based on CVRP, it is obvious that each customer served once by exactly one vehicle. The total length of the route and the total demand of the route must not exceed the constraint and ability of the vehicle respectively. Neighborhood selection method named pair-exchange is also used to improve the performance of CVRP by ex-change the position of an adjacent element. After a certain period, customers exchange their position and calculate fitness value, if the fitness value of the new solution is improved then the solution is accepted [19].

Quan-Ke Pana, M. F. Tasgetirenc, Yun-Chia L. [2007] proposed A discrete PSO ($DPSO_{VND}$) algorithm for the no-wait flow shop scheduling problem at chemical processing, food processing, concrete ware production and pharmaceutical processing. This method is used in scheduling to reduce makespan, total flow time and also computational time. In real-valued positions, standard PSO cannot generate discrete permutation so a new method used for discrete permutation which contains three components and those are the velocity of the particle, cognition part of the particle and social part of the particle. DPSO algorithm of No-wait flow shop scheduling problem uses the global neighborhood model. VND is a deterministic variant of PSO which is applied in PSO to improve the quality of solution for total flow time and makespan criteria. VND applied local search with DPSO to obtain a better quality of solution and consume less CPU time [20]. They also proposed a hybrid discrete particle swarm optimization (HDPSO) algorithm to reduce makespan of the no-wait flow shop problems with the criterion [21].

Table 2 present a summarized comparison among major variants of DPSO algorithm.

Algorithms	**Compare with Benchmark Function**	**Compare with Other Algorithms**	**Performance Analysis**
Fuzzy DPSO	Satisfy Burma14 and Berlin52 benchmark problems.	Better than original PSO but not better than Lin-Kernighan algorithm.	Effective in solving combinatorial problems.
DiMuPSO	Satisfy Sphere, Rosenbrock, Foxhole and Rastrigin benchmark function.	Outperforms traditional GA, PSO algorithms.	Very effective in finding optimal solutions to large problems with minimum iterations.
DPSO-SA	Satisfied	Outperform GA and Double population GA.	Feasible and effective approach for solving CVRP
$DPSO_{VND}$	Satisfied 110 and 31 benchmark instances.	Better than Elementary DPSO algorithm	Better quality result with less CPU runtime.
HDPSO	Satisfied Carlier, Revees, and Heller benchmark functions	Better than HPSO and $DPSO_{VND}$.	Robust and faster.

Table 2: Comparison among major DPSO variants.

5. Particle Swarm Optimization Algorithm for Dynamic Environments

Modified PSO algorithm for Dynamic Environments and its major variants and application explain in this section followed by their comparative analysis.

5.1. Modified PSO algorithm for Dynamic Environments

The original PSO algorithm cannot detect the change of goal value so the algorithm is influenced by its previous goal position memory. A small change in goal is self-correcting and the next fitness evaluation result will be closer to the new goal and swarms should follow and intersect the moving goal. Changes of the environment are effective for PSO to find the best position and choice of inertia parameter. PSO can dynamically track varying parabolic function. Tracking of nonlinear systems is effectively responding to a wide variety of changes in case of re-randomization of population changes is detected. Real-world problems are changed over time so that it can track the changes of the static object in a dynamic environment and compare how close it is with dynamically changed objectives [29]. Optimization method obtains its goal by handling changes in the dynamic system that detect the changes which actually occurred and react accurately to the change so that the optimum solution can be tracked properly. If the fitness value before and after iteration is different than change occurred. Recalculate the value after a change and Re-randomize the position. A detection method detects the changes and then activates the response method and this process will continue until the goal achieved. If the value of the fitness function does not match with the stored global best position, then we assume that a change of environment has occurred after that response method is initiated.

Algorithm 3: Modified PSO algorithm for Dynamic Environment:

Step 1: Initialize population *P*, Position *X,*

Step 2: Calculate the fitness value *F* of best position V_{lbest}

Step 3: Compare this best position with global best V_{gbest}. If $V_{lbest} > V_{gbest}$ then update the optimum value.

Step 4: Calculate the fitness value *F* of best position V_{pbest} after changes occurred in the environment.

Step 5: Compare the values of both after and before the change of environment and update with the best value.

Step 6: Repeat step 4 to step 5 after every iteration until it reaches the maximum iteration.

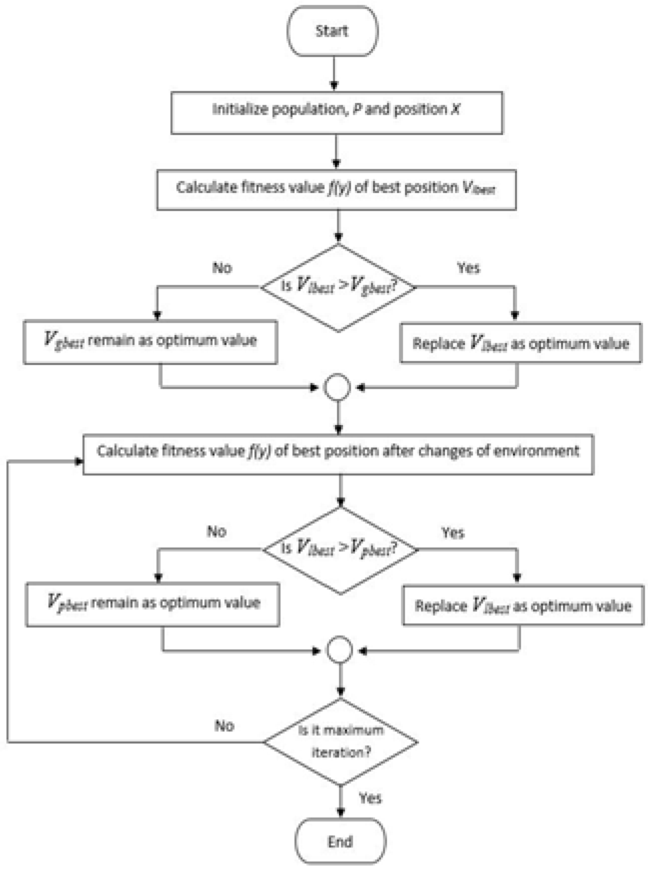

Figure 6: Flow chart of PSO algorithm for Dynamic Environments

5.2. Variants and Applications

Anthony Carlisle, Gerry Dozier [2000] proposed an adaptive PSO for dynamic environments. Particles change their position dynamically, If the movement of the goal occurred frequently then the next fitness evaluation value will be lower than current P vector (previous best) and for this particles cannot trace the moving goal. We tried to adjust the problem by replacing particles P vector with the current X vector (current best) and the particle forgot its previous position but forced to redefine their goal at that position. The goal moves through the constant (possibly zero) velocity in the search space for making the approach effective. When the fitness function is the distance between the particle and the goal and goal depends on the strength of the goal's signal [22].

R. C. Eberhart, Yuhui Shi [2001] proposed an algorithm (T-PSO_DE) for tracking and optimizing particle swarm in the dynamic environments to vary the location where optimum values occur. Due to the changes in the environment, most of the computation time used to reschedule the scheduling systems after each change. For tracking and optimizing dynamic systems first we have to achieve optimal values then evaluate the performance of swarms in the dynamic environment from the dynamic swarm's perspective. Whenever the environment is changed then the swarms reinitialize randomly and start finding a new best location based on velocity. Initializing with the old swarm is better for small changes where initializing with the random swarm is better for large changes [23]. They also proposed a fuzzy method to modify the inertia weight of the PSO algorithm dynamically [24].

A simple unimodal function can represent complex and non-linear real-world dynamic optimization problems. Morrison and De jong proposed a test function generator DF1 which can specify simple to complex dynamic environments. The goal of dynamic optimization is not only to find the optimum solution but also reach closer to search space over time [25].

S. Janson and M. Middendorf [2004] proposed a hierarchical approach of PSO for dynamic optimization problems. Hierarchical PSO (H-PSO) forms a tree of hierarchy that all nodes in the tree contain one particle. The particle with the best position so far places the upper side in the hierarchical tree. If a child particle contains the best position than the parent, then they can exchange their position. In the hierarchy tree, all inner nodes have the same out-degree but inner nodes of the deepest level have smaller out-degree and the difference is at most one. Hierarchical PSO cannot compute neighborhood due to fixed tree structure. Best position always contains the top position of the hierarchical tree [27].

J.J. Liang, P.N. Suganthan [2005] proposed a dynamic multi-swarm PSO where swarms are predefined or dynamically adjusted to the distance. New neighborhood topology is used in dynamic multi swarm PSO and the neighborhood structure has two characteristics: 1) Small Size Swarms which prefer a small number of particles for both simple and complex problems and good result use three to five particles. 2) Randomly Regrouping Schedule: Swarms of the small-sized group are searching using their current best information. Due to the convergence property of PSO, there is a high chance to converge in a local optimum. No information will exchange between swarms if neighborhood structure remains unchanged. Randomized regrouping schedule is introduced to avoid such kind of situation. Particles were reached in local optima by searching in a small group and after every R generation, they regrouped randomly and exchanged their best information with other swarms and continued searching until the global optima was found. DMS-PSO provides more freedom and better diversity to swarms because swarms are divided into many small groups and they can regroup randomly to exchange information with other swarms [28].

Comparison summary of few major variants of PSO algorithms for Dynamic Environments are present in Table 3.

Algorithms	**Compare with Benchmark Function**	**Compare with Other Algorithms**	**Performance Analysis**
APSO	Not compared	Overcome the changing goal detection problem of original PSO.	Cannot find any solution for V_g higher than 0.1
T-PSO_DE	Satisfied	Not compared	Track dynamically varying parabolic functions.
H-PSO	Satisfied Moving goal, Dynamic Rastrigin and Moving peaks benchmark function.	Better than original PSO and PH-PSO.	Perform better for multimodal function.
DMS-PSO	Satisfied six benchmark function and perform best for Rastrigin's function.	Better than PSO, PSO-cf, PSO-local, PSO-cf-local, UPSO, CPSO but in Rosenbrock's function FDR-PSO performs better.	Perform better in multimodal problems.

Table 3: Comparison table of PSO algorithm for Dynamic Environments

6. Particle Swarm Optimization Algorithm for Noisy Environments

In this section firstly we discuss about Modified PSO algorithm for Noisy Environments. Then Application and comparative analysis of few major variants of PSO algorithm for Noisy Environments.

6.1. Modified PSO algorithm for Noisy Environments

Due to noise, real-world problems become inaccurate and uncertain information that collapse performance of PSO algorithms for example, deviation and measurement errors. So it is difficult to identify the optimal solution by original PSO. For this reason, we need to use specialized PSO algorithm to handle such problems. Classical nonlinear programming techniques cannot solve all types of optimization problems due to the presence of multiple local and global optima. Evolutionary algorithms (Genetic Algorithm, Artificial life method) can quickly find the optima in complicated optimization problems than traditional methods through cooperation and competition between populations [29] [30]. Global optimization methods cannot detect the best optimal solution but it can detect sub-optimal solutions which are acceptable for a few problems but most of the problems desire optimal solution essentially. The traditional noisy function optimization method is simplex and polytope method which outperforms other optimization methods. The simplex method works without the assumption of continuity and local model of the function. For poor convergence properties, it faces difficulties and inefficient to work parallel but useful sequential applications. The performance of each particle measured by predefined fitness value where inertia weight is very important for convergence attribute which controls the influence of the previous velocity into current velocity. Addition of noise effective for the real-world problem where input contains noise. Sometimes noise helps PSO to avoid local optima. Due to the uncertainty of the noisy environment, the performance evaluation is only done by simulation of multiple sampling. At the same time, it is very difficult to find the global optima due to many local optima in the large search space. There are two types of noise which are adaptive and multiplicative noise. Multiplicative noise creates more corruption of fitness value than adaptive noise. The influence of noise cannot be ignored in high dimensional PSO problems that's why the performance of PSO decreases due to increase of noise. For the noisy environment, PSO uses a resampling method. The core idea of Equal Resampling PSO method is to maximize the probability of particles to an accurate selection of their best neighbors.

Algorithm 4: Modified PSO algorithm for noisy environment:

Step 1: Initialize population, P with random velocity, V and position X.

Step 2: Allocate T sampling budget by using Optimal Computing Budget Allocation (OCBA) to estimate the objective value and position of current swarm and all particles. Calculate pbest of each particle and gbest of the swarm.

Step 3: Update velocity and position of all particles

Step 4: Allocate T sampling budget by using OCBA for all particles with new positions estimating their objective values and update pbest of each particle.

Step 5: Use Hypothesis Test (HT) to form the new swarm.

Step 6: Order old position and new position of all particles from best to worst, and denote them $\theta_1, \theta_2, \ldots \theta_{2N}$. Let $\theta_1 = \theta_k$ and $\theta_2 = \theta_s$ and put θ_k into the next swarm.

Step 7: Perform HT for θ_s with θ_k which in the next swarm. If the null hypothesis holds then θ_s discarded otherwise θ_s put into the next swarm.

Step 8: If $k<N$ and $s<2N$ then go to step 7 else go to step 9.

Step 9: If $k=N$, that means a new swarm has been formed, else generate new particles randomly and evaluate them to form the new swarm.

Step 10: Update the gbest of the swarm if necessary.

Step 11: If it satisfies any predefined stopping condition then gbest is the result and its objective value. Else go to step 3 and continue.

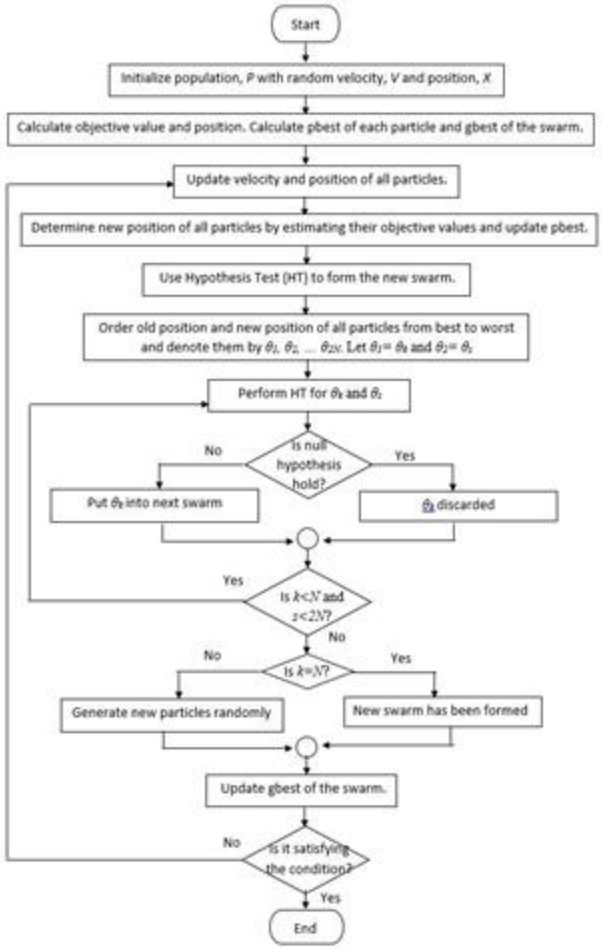

Figure 7: Flow chart of PSO algorithm for Noisy Environments

6.2. Variants and Application

Stefan Janson · Martin Middendorf [2006] proposed a new PSO method which combined with hierarchical PSO and Partitioned H-PSO for noisy and dynamic optimization problems. Partitioned hierarchical PSO (PH-PSO) is used to reduce the effect of noise in a dynamic environment. Controlling a time-varying system is really so difficult where information is corrupted by noise in a dynamic environment. In that case we have to distinguish background noise from dynamic environment. For small change of environment, it is beneficial to use top swarm as memory which stores good position before change occurred [26].

In standard PSO algorithm, addition of noise to optimization function have no vital instability problem rather this noise helps to overcome the local optima. multi-swarm PSO is particularly good for tracking some good local optima of dynamic function [32]. Like PH-PSO, multi-swarm PSO algorithms divided particles into different swarms called species independently track different optima in optimization function. This searching technique is not effective in noisy environments because in every iteration noisy functions initiate the response mechanism which will cancel the previous optimization best result. Noisy function improves the personal best position so that detection of better position is hampered. Hierarchy dependent re-evaluation strategy and hierarchy monitoring change detection methods are suitable for noisy and dynamic environments.

Hui Pan, Ling Wang, Bo Liu [2006] analyzed PSO algorithm for function optimization in noisy environment and proposed a hybrid PSO method named PHOOHT combined with hypothesis test and also use Optimal Computing Budget allocation (OCBA) for function optimization in noisy environment [32]. Hypothesis test is a statistical technique which tests hypotheses based on experiment data. Hypothesis test performed to select suitable solutions from multiple solutions. Two types of hypothesis: Null hypothesis H_0 and alternative hypothesis H_1. PSOOHT inherits basic population based searching techniques of PSO which reduce computational cost for finding good solution and the performance will improve if the number of simulations are determined from different solutions rather than equal simulation of all solutions. After OCBA based evaluation, HT stores the best solution and maintains diversity by deleting same valued particles to reduce repeated search [34].

Caifei Xiong, Zeyu Zhao [2018] proposed an opposition based PSO algorithm for noisy environments where Opposition based learning has been successfully implemented in differential evolution, reinforcement learning, neural network and PSO for increasing diversity of swarms. OBL helps some particles of PSO to overcome local optima in a noisy environment which is inspired from simultaneous consideration of an estimate and its corresponding opposite one. Due to noise, particles can be misled easily. For this we choose two worse particles and its opposite particles and select the fittest particle among them to reduce the influence of noise. OBL is used to increase diversity and helps to avoid premature convergence and select the top fittest particle from current and its opposite swarms. PSOGD can easily be trapped into local optima but OBL is able to overcome the local optima. OBL is used in PSOGD called PSOGD' which can avoid premature convergence and find 42/50 percent better solutions than PSOGD. Probabilistic opposition based learning used in PSO to deal with noisy optimization problems [35].

Junqi Zhang, Xixun Zhu, Yuheng Wang [2018] proposed a Duel environmental PSO which is applicable for both noisy and noiseless environments. Dual environmental PSO comes up with the concept of a weighted search center derived from top-k best particles in the swarms and path of the swarms decided from the current fitness value rather than the previous values. In a noise-free environment this method provides guidelines to avoid local optima. DEPSO can reduce noise more effectively than Traditional PSO in noisy environments. In a noise-free environment DEPSO can avoid local optimum and reach a better position. And for composition function DEPSO performs better than other algorithms with speed and accuracy. In a noisy environment DEPSO's non-historical weighted center guarantees its performance advantages. Traditional noise-free PSO algorithm lose their accuracy due to noise while DEPSO adapts this type of environment appropriately [36].

Table 4 Present comparative summary of few major variants of PSO algorithm for Noisy Environments.

Algorithms	**Compare with Benchmark Function**	**Compare with Other Algorithms**	**Performance Analysis**
PSO_NE_DE	Satisfied Rosenbrock, Levy No. 5 and Beale functions.	Not compared	Perform good results for Very scattered landscapes and multimodal functions.
PH-PSO	Satisfied for sphere, Rastrigin and Griewank function but not for Rosenbrock function	Better than the original PSO.	Perform better for unimodal function.
PSOOHT	Satisfied Goldstein Price, Branin, Hartman, Rastrigin and Shuber functions.	Better than PSO, PSOO, PSOHT.	Effective for deterministic problems but not effective for uncertain optimization problems.
PSOGD prime	Satisfied Elliptic, Rastrigin, Ackley, Schwefel and Rosenbrock functions.	Better PSOGD, PSO-OCBA, LAPSO, PSO-AN.	Avoid premature convergence of PSO.
DEPSO	Achieved 1^{st} position in 24 functions among 28 functions	Better than original PSO and noisy-free (CLPSO,ALPSO, IILPSO) and Noisy (PSOER, PSOERN, PSOLA)	Perform better for multimodal and composition functions.

Table 4: Comparison table of PSO algorithm for Noisy Environments

7. Hybrid Particle Swarm Optimization

Section 4 will discuss the applications of Hybrid PSO for different specialized problems space and their comparative analysis. Hybrid methods made by different PSO algorithms with other algorithms which are used to solve different optimization problems where original PSO methods failed to solve. Solution of Hybrid methods based on specific problem space. Specific version of PSO can find the solution of a specific problem but a combination of those algorithms can bring a solution to a new complex problem. Many PSO algorithms have few limitations to find optimal solutions, here we can use special methods to overcome those limitations and combine it with PSO methods. Many researchers modify and implement new hybrid algorithms for many complex real world problems.

7.1. Variants and Applications

Taher Niknam [2010] proposed a fuzzy adaptive hybrid PSO algorithm combined with nelder-mead algorithm for non-linear, non-smooth and non-convex Economic Dispatch (ED) problems. PSO can also solve the ED problem but it may have trapped into local optima due to dependency on parameters. For this reason, the proposed method is used to solve the ED problem with the value-point effect [37]. Inertia weight and learning factor of PSO adjust dynamically with fuzzy IF/THEN rules. This method uses FAPSO for optimization and NM algorithm for performing local search among the best solution which came from the FAPSO algorithm at every iteration to reach a global optimal solution. Proposed algorithm provides a better quality solution for non-linear, non-differentiable and discrete optimization problems. Apply NM search on this global position and select the local position then update parameters. Update velocity based on local and global position of FAPSO parameters. FAPSO-NM algorithm optimization method is very accurate and converges to global solution with short run time and small standard deviation [39].

Parham Moradi, Mozhgan Gholampour [2016] proposed a hybrid PSO (HPSO-LS) method for feature selection based on local search technique. Features selection was used to reduce dimensionality of datasets which choose a subset of salient features by eliminating irrelevant and redundant features. NP-Hard problem used to find optimal feature subset by enumerating and evaluating all possible subsets of features in the whole search space. It is computationally expensive to evaluate entire subsets of features so that heuristic or random search technique used to reduce computational time for finding optimal or sub-optimal subset of feature. Proposed hybrid method used to reduce computational time and better performance of both methods. To overcome feature selection problems in similar features, hybrid PSO integrates local search techniques (HPSO-LS) for feature selection using correlation information so that less similar features are selected with high probability. By specific subset size determination HPSO-LS reduces the number of salient features. HPSO-LS provides highest classification accuracy among PSO, GA, SA, ACO in feature selection. For low and high dimensional datasets HPSO-LS effectively removed irrelevant and redundant features [40].

A hybrid GPSO method proposed based on PSO and GA to classify the high dimensional microarray data [40]. Another hybrid method based on PSO and Support Vector Machine (SVM) have proposed to improve classification accuracy with small and appropriate feature subsets [42]. Catfish BPSO method used to improve BPSO performance of feature selection problems [43]. Two hybrid methods PSO-based relative redact and PSO based quick redact used in feature selection to increase efficiency.

Yannis Marinakis, Athanasios, Migdalas [2017] proposed a hybrid PSO algorithm with variable neighborhood search algorithm (VNS) for solving combinatorial optimization problems which are constrained shortest path problems. The minimum cost travelling paths between source to destination is called shortest path and only one minimum cost path is called constrained shortest path problem (CSP). For constrained shortest path problems, PSO algorithm was not appropriate for optimal solution due to loss of information. A new hybrid algorithm was proposed to speed up the process without loss of information and compared with inertia PSO and constriction PSO. This algorithm starts with local search neighborhood topology where neighborhoods are equal to 2 and increase every iteration until they become equal to the number of particles. Particles are moving for a number of iterations in a small swarm inside the whole swarm and then exchange the information easily to each other so that it increases the exploration abilities. In large search space variable neighborhood search (VNS) helps local neighborhood search to avoid being trapped into local optima and combine the neighborhood. Local search and VSN algorithm did not improve the result due to loss of information while transferring continuous to discrete PSO. That's why local search uses continuous value to avoid information losing [44].

Frank Jianga, Haiying Xiaa [2017] proposed a hybrid binary PSO which is combined with Wavelet Mutation to avoid premature conclusion of the best solution for solving continuous problems. Selecting a particle to move towards the search space using mutation in original PSO. This method will not return the best result all time due to use of fixed size mutation space. That's why the new hybrid PSO with wavelet method proposed to adjust mutation size dynamically. Velocity of binary particles updates by velocity vector. Compared with different benchmark functions, HPSOWM outperforms others hybrid PSO algorithms in finding optimal or sub-optimal solution and convergence rate in continuous search space. But convergence rate is not as fast as the other two algorithms. Solution quality and stability of HPSOWM is good due to its small mean and standard deviation. If the mutation size is large then, the algorithm fails to achieve improved stability and fine-tune ability. Then apply wavelet mutation function that can improve stability and fine-tune ability [45].

S. A. Hussein, A. A. Mahmood [2020] proposed a hybrid global local PSO method for human face recognition based on the Support Vector Machine (SVM). Support Vector Machine (SVM) can provide a global solution of facial recognition. The accuracy of SVM is dependent on trainee data. PSO is an effective technique to find the optimal solution in different domains such as video-based facial recognition or verification. PSO implementation with SVM to pursue ideal training parameters of face recognition system. The steps of facial recognition by GLAPSO-SVM are: firstly, read face image from database then, extract face feature by principal component analysis (PCA) method. Use the extract face feature is for training and testing GLAPSO-SVM model. GLAPSO proposed to overcome the choosing inertia weight parameter which recognized the value of gl-best and pl-best function for every particle's generation. Limitation of this method is to use fixed numbers for velocity coefficient which impacts on particles velocity performance. Another limitation is choosing inertia weight which is a user-supplied coefficient that provides balance between global and local solution. The lower value of inertia weight leads to convergence swarm's optima and higher value leads to investigate the whole search space. A method called AAPSO-SVM proposed to overcome those limitations [46].

Comparative summary of few hybrid PSO algorithms will discuss in Table 5.

Algorithms	**Compare with Benchmark Function**	**Compare with Other Algorithms**	**Performance Analysis**
FAPSO-NM	Not compared	Better than PSO, FAPSO	Require short run time and small standard deviation.
HPSO-LS	Compared with 12 datasets.	Better than GA, SA, ACO, HGAFS, HPSO-STS, SPSO-QR, CBPSO and PSO.	Feature selection with less computational time and higher accuracy.
PSOLGNT	Not compared	Better than ACO, RSVNS1, RSVNS2, SA, TS and GRASP.	Local search algorithms are most suitable for CSP problems.
BHPSOWM	Satisfied 18 benchmark function of three categories (Unimodal, Multimodal few local minima and many local minima)	Better than BGA and BPSO	Ensure good quality results with smaller runtime.
GLAPSO-SVM	Outperform CASIA-V5 and YALE-B human facial datasets.	Better than PSO-SVM, AAPSO-SVM	Assure better accuracy, less computational time and optimal training parameters.

Table 5: Comparison table of Hybrid PSO algorithms

8. Result Discussion

We have compared performance of different versions of PSO in terms of benchmark function and other related algorithms. BPSO perform better for both discrete and continuous problem space. DBPSO use to solve the floating point problem, Scheduling and routing problem of BPSO and also perform better than genetic algorithm (GA) where many global optima exist. NBPSO has stagnation problem that means velocity of the particle gradually become zero while the particle moving to the best position NBPSO trapped into local optima and converged prematurely. BPSO with Transform function (TF) overcome the problem of being trapped into local optima and slow convergence rate problem of PSO algorithm. SF-BPSO can provide better classification accuracy on feature selection of high dimensional datasets.

For Unit Commitment problem, PSO is better than GA in reduction of total generation cost in scheduling and also show better quality and convergence. Position matrix of fuzzy Discrete PSO shows the better solution of solving combinatorial optimization problems and outperforms original PSO algorithm but not better than Lin-Kernighan algorithm [46]. Diversity and dynamic decision of particles in the search space makes DiMuPSO better than other optimization algorithms in problems with large iterations. DPSO-SA can ensure the convergence to the global optimal solution by avoiding trapped into local optima. DPSOVND used to reduce total flow time cost and computational complexity of scheduling problems and better than GA and Double population GA. DPSOVND is better than Elementary DPSO algorithm for better quality result with less CPU runtime. HDPSO is better than HPSO and DPSOVND because of its robustness and faster result.

Adaptive PSO perform better when the entire environment changes in the same rate and insufficient for localized fluctuated environment. APSO cannot find any solution if goal velocity (Vg) higher than (0.1). H-PSO Perform better for multimodal functions and better than original PSO and PH-PSO. DMS-PSO perform randomize regrouping for information exchange which can avoid to converged into local optima. T-PSO_DE can track dynamically varying parabolic functions.

PSO_NE_DE perform good results for Very scattered landscapes and multimodal functions. PH-PSO perform better for unimodal function. PSOOHT which combined PSO with HT and OCBA have solved the deterministic problems but not effective for uncertain optimization problems of PSO. OBL helps PSO to overcome the premature convergence in noisy environment by choosing fittest particles among two particles and their opposite particles. PSOGD prime avoid premature convergence of PSO and better PSOGD, PSO-OCBA, LAPSO, PSO-AN. DEPSO used the concept of weighted search to find optimal solution in multiplicative noisy environment. DEPSO can also overcome local optima in a noise free environment.

FAPSO combined with NM algorithm and used to solve the Economic Dispatch problem and successfully find the global optima where PSO trapped into local optima within short runtime. HPSO-LS provides highest classification accuracy among PSO, GA, SA, ACO in feature selection. For low and high dimensional datasets HPSO-LS effectively removed irrelevant and redundant features. PSO cannot provide optimal value in CSP problems due to loss of information, to overcome this problem a Hybrid PSOLGHT algorithm was proposed. HPSOWM used to find the optimal and sub-optimal solution and convergence rate in continuous search space. GLAPSO-SVM used to overcome the choosing inertia weight limitation of PSO.

9. Conclusion

In this research we described the procedure and compared performance of different algorithms of five PSO variants which was designed to overcome various limitations of the PSO algorithm with respect to different environments. Original PSO algorithm cannot find the optimal solution of every problem space with higher accuracy and better computational efficiency. That's why for different problem spaces PSO combined with other methods and found the optimal solution. Binary version performs better for scheduling and feature selection and addition of Transform function (TF) overcome the problem of being trapped into local optima and slow convergence rate problem of PSO algorithm. Discrete version is robust and find faster solution in combinatorial problems and problems with large iterations. PSO for Dynamic Environments version can track dynamically varying parabolic functions and find better solution in multimodal functions. PSO for Noisy Environments version can perform better in both unimodal and multimodal problems and also overcome the premature convergence. Hybrid version is more reliable and faster because of its problem specific solution of problems. PSO is one of the fruitful population based metaheuristic algorithm which can work as the core idea behind the solution of many optimization problems.

Our future aim is to perform application based comparison among different variants of PSO algorithm and using these optimization techniques we will modify an algorithm for shortest path optimization problems..

References

[1] Wang, F., Zhang, H., & Zhou, A. (2021). A particle swarm optimization algorithm for mixed-variable optimization problems. Swarm and Evolutionary Computation, 60, 100808.

[2] James Kennedy. Swarm Intelligence. Handbook of Nature-Inspired and Innovative Computing (pp 187-219)

[3] E. Bonabeau, M. Dorigo, G. Theraulaz. Swarm Intelligence: From Natural to Artificial system. Oxford University Press, 1999.

[4] Tanzila Islam, Md. Ezharul Islam. An Analysis of Foraging and Echolocation Behavior of Swarm Intelligence Algorithms in Optimization: ACO, BCO and BA. International Journal of Intelligence Science. Vol.08 No.01(2018).

[5] J. Kennedy, R. C. Eberhart (1995). Particle Swarm Optimization. In Proceedings of the IEEE International Conference on neural network IV (PP. 1942-1948).

[6] SS Aote, MM Raghuwanshi, L Malik (2013). A brief review on particle swarm optimization: limitations & future directions. International Journal of Computer Engineering (IJCSE). Vol 2 No. 05.

[7] R. Poli, J. Kennedy, T. Blackwell (2007). Particle swarm optimization: An overview. Swarm Intelligence (2007) 1: 33–57.

[8] Russell C. Eberhart, Yuhui Shi (2001). Particle swarm optimization: developments, applications and resources. Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546). Vol 2.

[9] Kennedy and Eberhart [1997]. A discrete binary version of the particle swarm algorithm. IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation. IEEE Vol 5 Oct 1997.

[10] M. F. Taşgetiren and Yun-Chia Liang [2003]. A binary particle swarm optimization algorithm for lot sizing problem. Journal of Economic and Social Research 5 (2), 1-20.

[11] [M. A. Khanesar, M. Teshnehlab [2007]. A novel binary particle swarm optimization. 27th Mediterranean Conference on Control & Automation 2007. IEEE.

[12] S. Mirjalili, S.Z. Mohan Hashim [2011]. A Study of Different Transform Function for Binary Version of Particle Swarm Optimization. Proceedings of the 2011 International Conference on Genetic and Evolutionary Methods (GEM 11).

[13] SL. gupta, Anurag Sing, A. Iqbal [2019]. Big Data Classification Using Scale-Free Binary Particle Swarm Optimization. Harmony Search and Nature Inspired Optimization Algorithms (pp 1177-1187).

[14] Hongwei Cai, Xue Li, Chungang Xie, Kai Guo, Hui Liu, Chunjiang Liu [2019]. Area-to-point heat conduction enhancement using binary particle swarm optimization. Applied Thermal Engineering. Vol. 155, 5 June 2019, Pages 449-460.

[15] Zwe-Lee Gaing [2003]. Discrete particle swarm optimization algorithm for unit commitment. IEEE Power Engineering Society General Meeting (IEEE Cat. No.03CH37491) Vol 4 July 2003.

[16] Wei Pang, Kang-ping W. Chun-guang Z. Long-jiang D. [2004]. Fuzzy discrete particle swarm optimization for solving traveling salesman problem. The Fourth International Conference on Computer and Information Technology, 2004. CIT '04. IEEE.

[17] Wei Pang, Kang-ping W. Chun-guang Z. Long-jiang D. [2004]. Modified particle swarm optimization based on space transformation for solving traveling salesman problem. Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826). Vol 7 Aug 2004.

[18] B. Al kazemi, C.K. Mohan [2005]. Discrete Multi-Phase Particle Swarm Optimization. Information Processing with Evolutionary Algorithms (pp 305-327).

[19] CHEN Ai-ling, YANG Gen-ke, WU Zhi-ming [2006]. Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. Journal of Zhejiang University-SCIENCE A Vol 7, (pp 607–614)

[20] Quan-Ke Pana, M. F. Tasgetirenc, Yun-Chia L. [2008]. A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem. Computers & Operations Research. Vol 35, Issue 9, Sep 2008, (pp 2807-2839).

[21] Quan-Ke Pana, M. F. Tasgetirenc, Yun-Chia L. [2007]. A hybrid discrete particle swarm optimization algorithm for the no-wait flow shop scheduling problem with makespan criterion. The International Journal of Advanced Manufacturing Technology. Vol 38 (pp 337–347).

[22] Anthony Carlisle, Gerry Dozier [2000]. Adapting particle swarm optimization to dynamic environments. International conference on artificial intelligence, 2000.

[23] R. C. Eberhart, Yuhui Shi [2001]. Tracking and optimizing dynamic systems with particle swarms. Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546). Vol 2 May 2001.

[24] R. C. Eberhart, Yuhui Shi [2001]. Fuzzy adaptive particle swarm optimization. Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546). Vol 2 May 2001.

[25] R.W. Morrison, K.A. De Jong. A test problem generator for non-stationary environments. Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406). Vol 3.

[26] S. Janson and M. Middendorf [2006]. A hierarchical particle swarm optimizer for noisy and dynamic environments. Genetic Programming and Evolvable Machines (2006) Vol 7 (pp 329–354).

[27] S. Janson and M. Middendorf [2004]. A Hierarchical Particle Swarm Optimizer for Dynamic Optimization Problems. Applications of Evolutionary Computing (pp 513-524).

[28] J.J. Liang, P.N. Suganthan [2005]. Dynamic multi-swarm particle swarm optimizer. Proceedings IEEE Swarm Intelligence Symposium, 2005. SIS 2005.

[29] Xiaodong Li, Khanh Hoa Dam [2003]. Comparing particle swarms for tracking extrema in dynamic environments. The 2003 Congress on Evolutionary Computation, 2003. CEC '03. Vol 3.

[30] Darrell Whitley [1994]. A genetic algorithm tutorial. Statistics and Computing volume 4, pp (65–85).

[31] M. Mitchell & S. Forrest [1994]. Genetic Algorithms and Artificial Life. Artificial Life Vol 1, Issue 3 Spring 1994 (p.267-289).

[32] T.M. Blackwell & J. Branke [2004]. Multi-swarm optimization in dynamic environments. In Application of Evolutionary computing, LNCS 3005. pp 489-500.

[33] Hui Pan, Ling Wang, Bo Liu [2006]. Particle swarm optimization for function optimization in noisy environment. Applied Mathematics and Computation Vol 181, Issue 2, 15 Oct 2006, p. 908-919.

[34] J. Rada-Vilela, M. Zhang, M. Johnston [2013]. Optimal computing budget allocation in particle swarm optimization. GECCO '13. Proceedings of the 15th annual conference on Genetic and evolutionary computation. P. 81–88.

[35] Caifei Xiong, Zeyu Zhao [2018]. An opposition-based particle swarm optimization algorithm for noisy environments. IEEE 15th International Conference on Networking, Sensing and Control (ICNSC).

[36] Junqi Zhang, Xixun Zhu, Yuheng Wang [2018]. Dual-Environmental Particle Swarm Optimizer in Noisy and Noise-Free Environments. IEEE Transactions on Cybernetics. Vol 49 , Issue: 6 , June 2019.

[37] K. E. parsopoulos, M.N. Vrahatis [2001]. Particle swarm optimizer in noisy and continuously changing environments. Published: WWW.researchgate.net/publication/2493918.

[38] Taher Niknam, H. D. Mojarrad, M. Nayeripour [2010]. A new fuzzy adaptive particle swarm optimization for non-smooth economic dispatch. Energy Volume 35, Issue 4, P. (1764-1778).

[39] Taher Niknam [2010]. A new fuzzy adaptive hybrid particle swarm optimization algorithm for non-linear, non-smooth and non-convex economic dispatch problem. Applied Energy Vol 87, Issue 1, P. (327-339).

[40] P. Moradi, M. Gholampour [2016]. A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy. Applied Soft Computing Vol 43, P. (117-130).

[41] E-G. Talbi, L. Jourdan, J. Garcia-Nieto, E. Alba [2008]. Comparison of population based metaheuristics for feature selection: Application to microarray data classification. IEEE/ACS International Conference on Computer Systems and Applications. Pp. 45-52.

[42] S.-W. Lin, K.-C. Ying, S.-C. Chen, Z.-J. Lee [2008]. Particle swarm optimization for parameter determination and feature selection of support vector machines. Expert Systems with Applications Vol 35, Issue 4, P. (1817-1824).

[43] L.-Y. Chuang, S.-W. Tsai, C.-H. Yang [2011]. Improved binary particle swarm optimization using catfish effect for feature selection. Expert Systems with Applications Vol 38, Issue 10, P. (12699-12707)

[44] Y. Marinakisa, A. Migdalas, A. Sifaleras [2017]. A hybrid Particle Swarm Optimization – Variable Neighborhood Search algorithm for Constrained Shortest Path problems. European Journal of Operational Research. Vol 261, Issue 3, P. (819-834).

[45] Frank Jiang, Haiying Xiaa [2017]. A new binary hybrid particle swarm optimization with wavelet mutation. Knowledge-Based Systems. Vol 130, P. (90-101).

[46] S. A. Hussein, A. A. Mahmood [2020]. A Hybrid Global Local Adaptive Particle Swarm Optimization-Based Support Vector Machine Model for Human Facial Authentication. Journal of Southwest Jiaotong University. Vol 55, No 1.