



A quarter century of particle swarm optimization

Shi Cheng¹ · Hui Lu² · Xiujuan Lei¹ · Yuhui Shi³

Received: 4 January 2018 / Accepted: 22 March 2018 / Published online: 4 April 2018
© The Author(s) 2018

Abstract

Particle swarm optimization (PSO) is a population-based stochastic algorithm modeled on the social behaviors observed in flocking birds. Over the past quarter century, the particle swarm optimization algorithm has attracted many researchers' attention. Through the convergent operation and divergent operation, individuals in PSO group and diverge in the search space/objective space. In this paper, the historical development, the state-of-the-art, and the applications of the PSO algorithms are reviewed. In addition, the characteristics and issues of the PSO algorithm are also discussed from the evolution and learning perspectives. Every individual in the PSO algorithm learns from itself and another particle with good fitness value. The search performance and convergence speed were affected by different learning strategies. The scheduling and data-mining problems are illustrated as two typical cases of PSO algorithm solving real-world application problems. With the analysis of different evolution and learning strategies, particle swarm optimization algorithm could be utilized on solving more real-world application problems effectively, and the strength and limitation of various PSO algorithms could be revealed.

Keywords Particle swarm optimization · Developmental swarm intelligence · Population diversity · Exploration/exploitation

List of symbols

x_{id} Position of the i th particle at the d th dimension

v_{id} Velocity of the i th particle at the d th dimension
 p_{id} Personal best position of the i th particle at the d th dimension
 p_{nd} Neighborhood best position of the i th particle at the d th dimension
 $f(\mathbf{x}_i)$ Fitness value: the objective function value of \mathbf{x}_i
 t Iteration number
 maxFEs Maximum number of iteration
 u_d, l_d Upper/lower boundary of the d th search variable
 V_{\max} Maximum value of velocity
 S Population size: the number of particles in a population
 D Number of decision variables
 w Inertia weight
 c_1, c_2 Cognitive/social acceleration coefficient
 χ Constriction coefficient
 φ_1, φ_2 Acceleration coefficient limits

This work was supported in part by the National Natural Science Foundation of China under Grant 61672334, 61671041, 61773119, 61703256, and 61771297; in part by the Shenzhen Science and Technology Innovation Committee under grant number ZDSYS201703031748284; and in part by the Fundamental Research Funds for the Central Universities under Grant GK201703062.

✉ Hui Lu
mluhui@buaa.edu.cn
Shi Cheng
cheng@snnu.edu.cn
Xiujuan Lei
xjlei@snnu.edu.cn
Yuhui Shi
shiyh@sustc.edu.cn

- ¹ School of Computer Science, Shaanxi Normal University, Xi'an 710119, China
- ² School of Electronic and Information Engineering, Beihang University, Beijing 100191, China
- ³ Shenzhen Key Lab of Computational Intelligence, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China

Introduction

Many algorithms vanished after a decade. The particle swarm optimization algorithm is still attracted many researchers' attention over nearly a quarter century. Swarm intelligence, which is based on a population of individuals, is a collection of nature-inspired searching techniques. Particle swarm optimization (PSO), which is one of swarm intelligence algorithms, was invented by Eberhart and Kennedy in 1995 [29,42]. It is a population-based stochastic algorithm modeled on social behaviors observed in flocking birds. Each particle, which represents a solution, flies through the search space with a velocity that is dynamically adjusted according to its own and its companion's historical behaviors. The particles tend to fly toward better search areas over the course of the search process [31].

Optimization, in general, is concerned with finding “best available” solution(s) for a given problem. For optimization problems, it can be simply divided into unimodal problem and multimodal problem. As the name indicated, unimodal problem has only one optimum solution, on the contrary, multi-modal functions have several or numerous optimum solutions, of which many are local optimal solutions. It is difficult for optimization algorithms to find the global optimum solutions. Avoiding premature converge is important in multimodal problem optimization, i.e., an algorithm should have a balance between fast converge speed and the ability of “jumping out” of local optima.

Many real-world applications could be modeled as optimization problems. As an outstanding swarm intelligence algorithm, the particle swarm optimization has been widely used to solve enormous real-world problems. It is difficult, if not impossible, to list all the problems that could be solved via the PSO algorithms. The scheduling problem and data mining problem are two typical real-world applications that could be solved by the PSO algorithms.

The aim of this paper is to provide a comprehensive review of the particle swarm optimization algorithms. The remaining of the paper is organized as follows. The basic concepts and the developmental history of PSO algorithm are reviewed in Sect. 2. In Sect. 3, different variants of PSO algorithms and the applications on solving various problems are introduced. The characteristics and issues of PSO algorithms are described in Sect. 4. Section 5 gives some real-world application of PSO algorithms. Finally, Sects. 6 and 7 concludes with future research directions and some remarks, respectively.

The historical development

Rome was not built in a day. After a quarter century, several papers on the PSO algorithms, which include the original and canonical PSO algorithms, have been cited more than

ten thousand times [29,42,71]. There has been a great development for PSO algorithm after it was proposed in 1995 [29,42].

Original particle swarm optimization

Each particle represents a potential solution in particle swarm optimization, and this solution is a point in the n -dimensional solution space. The original PSO algorithm is simple in concept and easy in implementation [29,42]. The velocity v_{ij} and position x_{ij} of the j th dimension of the i th particle are updated as follow [32,43]:

$$v_{ij} = v_{ij} + c_1 \text{rand}() (p_{ij} - x_{ij}) + c_2 \text{Rand}() (p_{nj} - x_{ij}) \quad (1)$$

$$x_{ij} = x_{ij} + v_{ij} \quad (2)$$

where c_1 and c_2 are positive constants, and $\text{rand}()$ and $\text{Rand}()$ are two random functions in the range $[0, 1)$ and are different for each dimension and each particle.

Canonical particle swarm optimization

In the original particle swarm optimizer, the velocity is difficult to control during the search process. The final solution is heavily dependent on the initial seeds (population). For different problems, there should be different balances between the local search ability and global search ability. Shi and Eberhart introduced a new parameter, an inertia weight w , to balance the exploration and exploitation [71,78]. This inertia weight w is added to Eq. (1), and it can be a constant, linear decreasing value over time [72], or fuzzy value [73,76]. The new velocity update equation is as follows

$$v_{ij} = wv_{ij} + c_1 \text{rand}() (p_{ij} - x_{ij}) + c_2 \text{Rand}() (p_{nj} - x_{ij}) \quad (3)$$

Adding an inertia weight w can increase the probability for an algorithm to converge to better solutions, and have a way to control the whole process of algorithm's searching. Generally speaking, algorithm should have a more exploration and less exploitation ability at first, which has a high probability to find more local optima. Exploration should be decreased, and exploitation should be increased to refine candidate solutions over the time. Accordingly, the inertia weight w , should be linearly decreased or even dynamically determined by a fuzzy system.

The equations of classical particle swarm optimization algorithm can be rewritten in the vector form. The velocity and position update equations are as follow [32,43,71]:

$$\mathbf{v}_i \leftarrow w\mathbf{v}_i + c_1 \mathbf{rand}() (\mathbf{p}_i - \mathbf{x}_i) + c_2 \mathbf{Rand}() (\mathbf{p}_n - \mathbf{x}_i) \quad (4)$$

$$\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}_i \quad (5)$$

where w denotes the inertia weight [73,78], c_1 and c_2 are two positive acceleration constants, $\text{rand}()$ and $\text{Rand}()$ are two random functions to generate uniformly distributed random numbers in the range $[0, 1)$ and are different for each dimension and each particle, \mathbf{x}_i represents the i th particle’s position, \mathbf{v}_i represents the i th particle’s velocity, \mathbf{p}_i is termed as personal best, which refers to the best position found by the i th particle, and \mathbf{p}_g is termed as local best, which refers to the position found by the members in the i th particle’s neighborhood that has the best fitness evaluation value so far.

The inertia weight w can be different for different particle at different dimension. The inertia weight can be written as \mathbf{w}_i . Consider the iteration number t , the equations are rewritten as:

$$\begin{aligned} \mathbf{v}_i(t + 1) &\leftarrow \mathbf{w}_i \mathbf{v}_i(t) + c_1 \text{rand}() (\mathbf{p}_i - \mathbf{x}_i(t)) \\ &\quad + c_2 \text{Rand}() (\mathbf{p}_n - \mathbf{x}_i(t)) \\ \mathbf{x}_i(t + 1) &\leftarrow \mathbf{x}_i(t) + \mathbf{v}_i(t + 1) \end{aligned}$$

Random variables are frequently utilized in swarm optimization algorithms. The length of search step is not determined in the optimization. This approach belongs to an interesting class of algorithms that are known as randomized algorithms. A randomized algorithm does not guarantee an exact result but instead provides a high probability guarantee that it will return the correct answer or one close to it. The result(s) of optimization may be different in each run, but the algorithm has a high probability to find a “good enough” solution(s).

The flowchart of particle swarm optimization algorithm is shown in Fig. 1 and the procedure of particle swarm optimization algorithm is given in Algorithm 1.

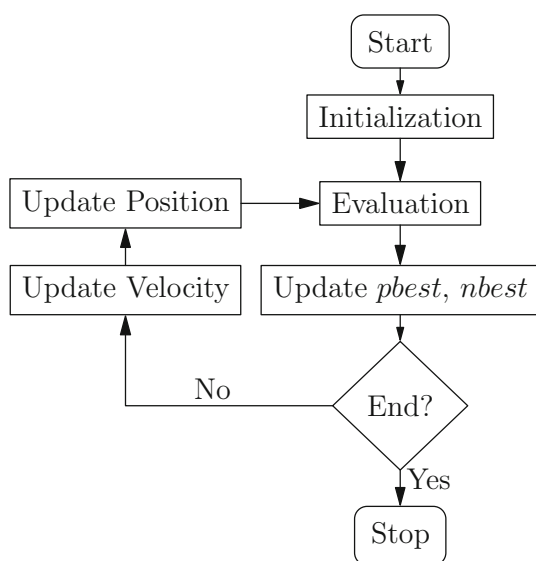


Fig. 1 The flowchart of particle swarm optimization algorithm

Algorithm 1: The procedure of particle swarm optimization algorithm

```

1 Initialization: Initialize velocity and position randomly for each
  particle in every dimension;
2 while not find the “good” solution or not reach the maximum
  iteration do
3   Calculate each particle’s fitness value;
4   Compare fitness value between current value and best
  position in history (personal best, termed as pbest). For each
  particle, if fitness value of current position is better than
  pbest, then update pbest as current position;
5   Selection of a particle which has the best fitness value from
  current particle’s neighborhood, this particle is called the
  neighborhood best (termed as nbest)1;
6   for each particle do
7     Update particle’s velocity according equation (4);
8     Update particle’s position according equation (5);
    
```

A particle updates its velocity according to Eq. (4), and updates its position according to Eq. (5). The $c_1 \text{rand}() (\mathbf{p}_i - \mathbf{x}_i)$ part can be seen as a cognitive behavior, while $c_2 \text{Rand}() (\mathbf{p}_g - \mathbf{x}_i)$ part can be seen as a social behavior.

In particle swarm optimization, a particle not only learns from its own experience, but also learns from its companions. It indicates that a particle’s “moving position” is determined by its own experience and its neighbors’ experience.

Fully informed particle swarm optimization

Fully informed PSO (FIPS) does not share the concept of “global/local best”. A particle in FIPS does not follow the leader in its neighborhood, but follow all other particles in its neighborhood. The basic equations of the FIPS algorithm are as follow [44,56]:

$$\mathbf{v}_i \leftarrow \chi \left(\mathbf{v}_i + \sum_{k=1}^{N_i} \frac{U(0, \varphi) (\mathbf{p}_{\text{nbr}(k)} - \mathbf{x}_i)}{N_i} \right) \tag{6}$$

$$\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}_i \tag{7}$$

where χ denotes the acceleration coefficient, $U(0, \varphi)$ is a random function to generate random numbers in the range $[0, \varphi]$, N_i represents the neighborhood size of the i th particle, and $\mathbf{p}_{\text{nbr}(k)}$ represents the k th particle’s personal best position. Each particle in PSO algorithm represents a potential solution which is a point in the D -dimensional solution space. Each particle is associated with two vectors, i.e., the velocity vector and the position vector. Throughout this paper, i is used to index the particles or solutions (from 1 to S) and d is used to index the dimensions (from 1 to D). The S represents the

¹ If the current particle’s neighborhood includes all particles then this neighborhood best is the global best (termed as *gbest*), otherwise, it is the local best (termed as *lbest*).

number of particles, and D represents the number of dimensions. The position of the i th particle is represented as \mathbf{x}_i , $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{id}, \dots, x_{iD}]$. x_{id} represents the value of the d th dimension for the i th solution, where $i = 1, 2, \dots, S$, and $d = 1, 2, \dots, D$. The velocity of a particle is represented as \mathbf{v}_i , $\mathbf{v}_i = [v_{i1}, v_{i2}, \dots, v_{id}, \dots, v_{iD}]$.

The parameter w was introduced to control the global search and local search ability [71]. The PSO algorithm with the inertia weight is termed as the canonical/classical PSO. The canonical PSO algorithm could be rewritten as the PSO with constricted factor (PSO-CF) version [26]. The explosion, stability, and convergence of PSO-CF algorithm were analyzed via the new velocity updating equation.

$$p_{md} = \frac{\varphi_1 p_{id} + \varphi_2 p_{nd}}{\varphi_1 + \varphi_2} \quad (8)$$

Equation (4) is reformed as follows:

$$v_{id}^{t+1} = \chi(v_{id}^t + \varphi \text{rand}() (p_{md}^t - x_{id}^t)) \quad (9)$$

where the $\varphi = \varphi_1 + \varphi_2$. Equation (9) also could be reformed as follows:

$$v_{id}^{t+1} = \chi[v_{id}^t + \varphi_1 \text{rand}() (p_{id}^t - x_{id}^t) + \varphi_2 \text{rand}() (p_{nd}^t - x_{id}^t)] \quad (10)$$

Based on Eqs. (10), (9) could be easily transferred to Eq. (4) via $\chi = w$, $c_1 = \chi \times \varphi_1$, and $c_2 = \chi \times \varphi_2$. The inertia weights and constriction factors in PSO were discussed in [30].

The state-of-the-art

There are many variants of PSO algorithms which have been proposed, including multiple swarms, new efficient learning strategy, diversity maintaining strategy, and hybrid algorithms to solve various optimization problems.

Algorithms

It's meaningless and difficult, if not impossible, to count the number of PSO variants. The following list gives the name of several PSO variants for examples. These PSO variants could be generally categorized into five groups:

1. The first adjusts the configuration parameters to balance the global and local search abilities:
 - Standard particle swarm optimization (SPSO-BK) [8],
 - Standard particle swarm optimization 2011 [24],

- Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients (HPSO-TVAC) [66],
 - Adaptive particle swarm optimization (APSO) [96].
2. The second aims to enhance the population diversity by designing new information propagation strategies:
 - Fully informed particle swarm optimization [44,56],
 - Particle swarm optimization algorithm with Re-initialize strategy [20].
 3. The third is the hybridization of PSO algorithm and other auxiliary search techniques:
 - Multiple strategies-based orthogonal design particle swarm optimizer (MSODPSO) [64],
 - Particle swarm optimization algorithm with parasitic behavior (PSOPB) [62],
 - Particle swarm optimization with dynamical exploitation space reduction strategy (DESP-PSO) [21],
 - Particle swarm optimization with an aging leader and challengers (ALC-PSO) [13],
 - Adaptive particle swarm optimization with heterogeneous multicore parallelism and GPU acceleration [85].
 4. The fourth introduces multiple swarms or coevolving groups to improve the global search ability:
 - Cooperative particle swarm optimizer (CPSO) [7],
 - Dynamic multi-swarm particle swarm optimizer (DMS-PSO) [50],
 - Particle swarm optimization with interswarm interactive learning strategy [63],
 - Cooperatively coevolving particle swarms (CCPSO2) [48].
 5. The fifth is PSO algorithms with new efficient learning strategy:
 - Comprehensive learning particle swarm optimizer (CLPSO) [49],
 - Orthogonal particle swarm optimization (OPSO) [38],
 - Orthogonal learning particle swarm optimization (OLPSO) [97],
 - Genetic learning particle swarm optimization [35].

Different topology structure can be utilized in PSO, which will have different strategies to share search information for every particle. Global star and local ring are two most commonly used structures. A PSO with global star structure, where all particles are connected to each other, has the smallest average distance in swarm, and on the contrary, a PSO with local ring structure, where every particle is connected to

two near particles, has the biggest average distance in swarm [22,56].

Problems

The optimization problem could be classified into several categories. According to the number of objectives, the optimization problems can be divided as single-objective, multi-objective problems, and many-objective problems. Based on the properties of decision variables, the problems are labeled as dynamic problems, large-scale optimization problems, *etc.*

Single-objective problems

The single-objective problems is the basic problems for optimization. The original PSO algorithm was tested on single-objective problems with continuous search ranges [29,42]. The modified PSO algorithm has been used to solve discrete optimization or combinatorial optimization problems [6].

Multiojective problems

Multiojective Optimization refers to optimization problems that involve two or three conflicting objectives, and a set of solutions is sought instead of one [27]. For the multi-objective problems, the traditional mathematical programming techniques have to perform a series of separate runs to satisfy different objectives [27]. Many kinds of PSO algorithms have been used to solving multiojective optimization problems [28], such as adaptive multiojective PSO algorithm [36], geometric structure-based PSO algorithm [94], normalized ranking based PSO algorithm [17], just to name a few. An essential issue in utilizing particle swarm optimizer to solve multiojective or many-objective problems is the setting of the personal best p_i and the neighborhood best p_n [17].

Many-objective problems

Many-objective optimization refers to algorithms which solve problems with more than three conflicting objectives [33]. Unlike the multiojective optimization, the Pareto optimality is not effective because nearly all solutions are Pareto non-dominated for problems with more than three objectives [86]. The population diversity is another difficulty for many objective optimization because the similarity is hard to estimate in high-dimensional space [86]. The solution comparison or the selection of more “representative” solutions is an essential issue in many-objective optimization. The various PSO algorithms have been proposed to solve many-objective optimization problems, such as PSO algorithm with two-stage strategy and a parallel cell coordinate system [39],

the normalized ranking-based PSO algorithm [17], or two archive algorithm [86], just to name a few.

Multimodal multiobjective problems

PSO algorithm is used to solve new kind of optimization problems, such as multimodal multiobjective problems [95]. For multimodal multiobjective problems, the algorithm needs to find multiple global optima in search space which satisfy more than one objective in objective space [19].

Beside the aforementioned problems, PSO algorithms also have been applied to other problems, such as large scale problems [15,16], dynamic multimodal optimization problems [88], *etc.*

Characteristics and issues

The evolution and learning are two basic characteristics of PSO algorithms, or more general, swarm intelligence algorithms. There are many unsolved issues for swarm intelligence algorithms, such as the exploration versus exploitation, the population diversity, and the parameter setting.

Evolution

Evolution is an important concept in evolutionary computation and swarm intelligence. In biology, the “evolution” is defined as “a change in the heritable characteristics of biological populations over successive generations.” In swarm intelligence algorithms, the evolution indicates that the solutions are generated iteration by iteration, and move toward better and better search areas over the course of the search process. For PSO algorithms, the “leader” particles or the learned particles are selected in this self-evolution or self-improvement process. The obtained fitness values are improved based on the self-evolution process. Normally, there is one group in the swarm intelligence algorithms, and information is propagated in all individuals. The hybrid algorithm and multiple sub-swarms are two special swarms which have different evolution methods. For these two kinds of algorithms, the search information is different for each group and the search information is exchanged at certain times.

Hybrid algorithms

The aim of hybrid algorithm is to combine the strength of two or more algorithms, while simultaneously trying to minimize any substantial disadvantage [81]. The PSO algorithm has been combined with the fuzzy modeling [83], self-organizing radial basis function (RBF) neural network [37] to solve various problems.

Multiple sub-swarm

Niching method is able to locate multiple solutions in multi-modal search areas. The PSO algorithm could be combined with niching techniques to rapidly solve problems [9]. The swarm of particles could be divided into multiple sub-swarms [93], dynamically changing group size [48], or multiple species [99]. The different swarm could have different function during the search, and the search information could be propagated effectively via interswarm interactive learning strategy [63].

Learning

Learning has two aspects in particle swarm optimization algorithm. One is learning from the problems, which means the algorithm is able to adjust its search strategies or parameters dynamically during the search. The other one is learning from particles themselves, which indicates the approach of search information propagated among all particles.

The developmental swarm intelligence (DSI) is a new framework of swarm intelligence [77]. DSI algorithm has two kinds of functionalities: capability learning and capacity developing. The capacity developing is a top-level learning or macro-level learning methodology. The capacity developing describes the learning ability of an algorithm to adaptively change its parameters, structures, and/or its learning potential according to the search states of the problem to be solved. In other words, the capacity developing is the search potential possessed by an algorithm. The capability learning is a bottom-level learning or micro-level learning. The capability learning describes the ability for an algorithm to find better solution(s) from current solution(s) with the learning capacity it possesses. The flowchart of developmental particle swarm optimization algorithm is shown in Fig. 2.

Learning from problem

The capability learning indicates that the algorithm has an ability that could learning from problem. The PSO could dynamically adjust its search strategy during the search. Several particle swarm optimization algorithms with different capability learning strategies have been proposed, such as adaptive PSO [96].

The objects of learning

The objects of learning could be different for various PSO algorithms. For example, a particle is able to learn personal best search information from other particles in the comprehensive learning PSO (CLPSO) [49]. The learning equation of CLPSO is given in Eq. (11).

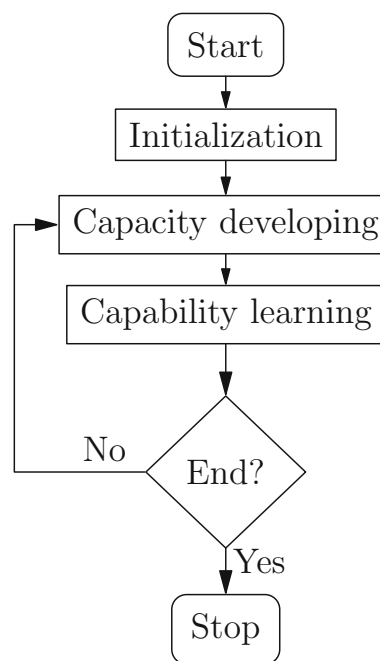


Fig. 2 The flowchart of developmental particle swarm optimization algorithm

$$v_{id} = wv_{id} + c \times \text{rand}() (pbest_{fi(d)d} - x_{id}) \quad (11)$$

where $\mathbf{f}_i = [f_{i(1)}, f_{i(2)}, \dots, f_{i(D)}]$ defines which particle's $pbest$ the i th particle will follow.

The speed of learning

A particle updates its position in the search space at each iteration. The velocity update equation consists of three parts, which are previous velocity, cognitive part, and social part. The cognitive part means that a particle learns from its own searching experience, and correspondingly, the social part means that a particle can learn from other particles, or learn from the best in its neighbors in particular. Topology defines the neighborhood of a particle [22].

Particle swarm optimization algorithm has different kinds of topology structures, e.g., star, ring, four clusters, or Von Neumann structure. A particle in a PSO with a different structure has different number of particles in its neighborhood with a different scope. Learning from a different neighbor means that a particle follows different neighborhood (or local) best, in other words, topology structure determines the connections among particles, and the strategy of search information propagation.

In general, PSO with star topology has the smallest diameter and average distance, which means that search information has the fastest propagation in all topologies, and on the contrary, PSO with ring topology has the largest diameter and average distance.

Exploration versus exploitation

The most important factor affecting an optimization algorithm's performance is its ability of "exploration" and "exploitation". Exploration means the ability of a search algorithm to explore different areas of the search space to have high probability to find good optimum. Exploitation, on the other hand, means the ability to concentrate the search around a promising region to refine a candidate solution.

A good optimization algorithm should optimally balance the two conflicted objectives, which indicates that the ability of exploration and ability of exploitation should be adjusted via the population diversity analysis when solving different problems or on different search stages. For example, to solve multimodal problem, great exploration ability means that algorithm has great possibility to "jump out" of local optima.

Population diversity

Population diversity is a measurement of population state in exploration or exploitation. It illustrates the information of particles' position, velocity, and cognitive. Particles getting diverge means that algorithm in an exploration state, on the contrary, particles converging into a small search area means that algorithm in an exploitation state.

Population diversity of PSO is useful for measuring and dynamically adjusting algorithm's ability of exploration or exploitation accordingly. Shi and Eberhart gave three definitions on population diversity, which are position diversity, velocity diversity, and cognitive diversity [74,75]. Position, velocity, and cognitive diversity are used to measure the distribution of particles' current positions, current velocities, and *pbests* (the best position found so far for each particles), respectively. From diversity measurements, the useful information can be obtained.

Low diversity, which particles converging into a small search area, is often regarded as the main cause of premature convergence. Several mechanisms have been proposed to promote diversity in particle swarm optimization, such as PSO with elitist re-initialization.

Parameter setting

In swarm intelligence research, one comment often received for any new proposed algorithm is "the authors should give a fair comparison on all algorithms in the paper." It's very normal that reviewers ask for a fair comparison. One reason from a real comment is as follows: "The thing is that the authors have certainly very carefully tuned their parameter values to get the best possible results on their test functions. However, I am almost certain that they did not do the same for the other methods they compared to." What is a fair compar-

ison among all algorithms? Usually, the proposed algorithm and the other compared algorithms are tested on a set of new benchmark functions, which are different from benchmark functions used by other algorithms when they were firstly proposed. Should all algorithms have different parameter settings, and each algorithm have exactly the same settings with the settings when it was firstly proposed? Or should all algorithms have the same parameter settings?

In the algorithm comparison, it maybe a good option for new variant of PSO algorithm or other swarm intelligence algorithms that compare the proposed algorithm with the standard PSO algorithm. It should be noted that there are two variants called standard particle swarm optimization (SPSO) algorithms. The first one, which termed as SPSO-BK, was defined by Bratton and Kennedy in 2007 [8], and the other one, which termed as SPSO-C, was defined by Clerc in 2006, 2007, and 2011 [25]. The analysis of these two algorithms was given in [23]. The strategy of population size setting is different for problems with different scale [12].

Real-world applications

Particle swarm optimization as one of outstanding swarm intelligent algorithms has been widely used to solve enormous real-world problems, such as optimal design of electric machines problem [47], Wi-Fi indoor positioning problem [14], indoor high precision three-dimensional positioning problem [10], energy management problem [57], economic load dispatch problems [61], just to name a few. It is no exaggeration to say that PSO has unique performance nearly in every area, like industrial engineering, intelligent manufacture, data mining, information and communication system, automatic control system, image processing. We cannot list all these applications because of the various areas. Here, we take scheduling problem and data mining problem as example to review the application of PSO in real-world problems.

Scheduling problem

Scheduling problem is one kind of combinatorial optimization problem and a very popular area in different industrial field. The job shop scheduling problem (JSP) [90], the test task scheduling problem (TTSP) [52], the parallel machine scheduling problem (PMSP) [91] are typical representatives of the scheduling problem. What they all have in common is the rational allocation of jobs or tasks to machines or resources. Therefore, we take the TTSP, the unrelated parallel machine scheduling problem (UPMSP) [91] and the flexible job shop scheduling problem (FJSP) [70] into consideration to illustrate the characteristics of these scheduling problem and the application of PSO.

TTSP is one of the key technologies to improve the performance of automatic test system (ATS) [82]. According to the actual problem, the mathematical model of TTSP is that test tasks have to be arranged on test resources. Each task may have multiple options to choose, and each task may be tested on multiple instruments at the same time. Although FJSP has similarity with TTSP, there are some difference between them. Each operation in FJSP has to be carried on only one machine and the operations of a job are in accordance with the predetermined order [53]. For UPMSPP each job requires a given processing time and machines are considered unrelated when the processing times of the jobs depend on the machine to which they are assigned to [84]. In these mathematical models, each job of FJSP and PMSP can be performed on every machine with no constraint. However, a task in TTSP has to be performed on some predetermined resources.

PSO and various variant PSO or hybrid PSO have demonstrated their performance in solving these scheduling problems.

Test task scheduling problem (TTSP)

For TTSP, Lu proposed a hybrid particle swarm optimization and tabu search for single-objective TTSP with constraints [51]. PSO is used for solving the test task sequence problem and tabu search is used for the instrument resource dispatching problem. It is one kind of non-integrated strategy for solving the scheduling problem. A new kind of inertia weight related with the iteration process and a constraint handling mechanism based on coding strategy were used. An encoding strategy of every particle was invented for handling the serial task sequence constraints. Lu also combine PSO with variable neighborhood MOEA/D to solving multi-objective TTSP, which used PSO to find the ideal point in multi-objective evolutionary algorithm based on decomposition (MOEA/D) [54]. The makespan and the mean load of instrument are the two objectives. In addition, PSO was used as an embedded algorithm in an integrated solution framework based on packet scheduling and dispatching rule for job-based scheduling problems. PSO demonstrated its performance through comparison with other kind of meta-heuristics algorithms.

Flexible job shop scheduling problem (FJSP)

For FJSP, Nouiri investigated a two stage particle swarm optimization (2S-PSO), which consists of PSO after initial swarm for objective of makespan and PSO after final swarm for stability or other objective, to solve the flexible job shop predictive scheduling problem considering possible machine breakdowns [59]. The objective is to solve the problem under uncertainty with only one breakdown. 2S-PSO are

tested on various benchmark data varying from partial FJSP to total FJSP. The proposed 2S-PSO evaluates the effect of disruptions on the solution using the robustness and stability measure. Singh proposed an quantum behaved particle swarm optimization (QPSO) for FJSP [79]. QPSO can effectively address the drawback of PSO, which is easy to trap at local optimum due to the large reduction in velocity value as iteration proceeds and poses difficulty in reaching at best solution. In addition, mutation has been introduced in QPSO for avoiding the premature convergence. Zhang proposed an effective hybrid particle swarm optimization algorithm for multi-objective FJSP [98]. PSO and a tabu search algorithm are combined to obtain the local and global searching ability. It is a very useful integrated strategy for multi-objective optimization problems.

Parallel machine scheduling problem (PMSP)

For PMSP, Hulett focused on scheduling non-identical parallel batch processing machines to minimize total weighted tardiness and PSO was used to solve the problem [40]. The smallest position value rule is used to convert the continuous position values of the particle to a discrete job permutation. It is one kind of application for testing printed circuit boards in an electronics manufacturing facility. Likewise, a heuristic is proposed to simultaneously group the jobs into batches and schedule them on a machine. Shahidi-Zadeh investigated a comparison study for solving a bi-objective unrelated parallel batch processing machines scheduling problem [68]. The multi-objective particle swarm optimization (MOPSO), non-dominated sorting genetic algorithm (NSGA-II), multi-objective ant colony optimization algorithm (MOACO), and multi-objective harmony search (MOHS) are used to solve the problem. The MOPSO got a good performance in diversity and spacing of Pareto optimal frontiers. Shahvari focused on a bi-objective batch processing problem with dual-resources on unrelated-parallel machines. Four bi-objective PSO-based search algorithms are proposed to efficiently solve the optimization problem for medium- and large-size instances [69].

Data mining problem

Data mining problem has different branches, like outlier detection, association rule, cluster, classification, prediction. PSO can be used to solve all these branches. Therefore, we have various variant PSO, such as PSO for outlier detection, PSO for classification, PSO for association rule mining and PSO for prediction analysis for time series. In addition, these variant PSO has been used in sensor networks, medical dialysis, network security, financial monitoring, image processing and other fields.

Outlier detection

An outlier is a data which is different from the other data in that domain. This abnormal data or point can be very useful to describe the abnormality of that system. The outlier detection is useful in many applications [3,65]. Misinem proposed a rough set outlier detection strategy based on PSO, which is to find minimum non-reduct [58]. Ye proposed a new algorithm for high-dimensional outlier detection based on constrained PSO [92]. The concept of outliers is defined as sparsely populated patterns in lower-dimensional subspaces. PSO is used with a specifically designed particle coding and conversion strategy as well as some dimensionality-preserving updating techniques to the search for best abnormally sparse subspaces. Condition based maintenance (CBM) is gaining importance in industry because of the need to increase machine availability. An application of PSO is presented for detection of machinery fault for CBM [67]. It is also one kind of application for outlier detection. Alam used PSO-based clustering strategy to realize web bots detection [2]. Feng proposed a multi-objective vector evaluated PSO with time variant coefficients for outlier identification in power systems [34]. It is one kind of unsupervised classification of electric load data.

Association rule mining

Association rules aims in extracting important correlation, frequent pattern, association or casuals structures among the set of items in the data sets [5]. Association rule basically extracts the patterns from the database based on the two measures such as minimum support and minimum confidence. Ankita had reviewed the application of PSO in association rule mining [5]. PSO is implemented for association rule mining in two ways. One is to generate rules by implementing PSO in the traditional algorithm of association rule mining. Another is optimization of association rule generated by traditional algorithm using PSO. Maragatham investigated a weighted particle swarm optimization technique for optimizing association rules [55]. They consider the utility based temporal association rule mining method for generating the association rules and PSO is used to optimize the generated rules. Indira proposed an adaptive PSO that yields a finer solution by performing a diversified search over the entire search space [41]. The parameters such as inertia weight and acceleration coefficients are adjusted dynamically. We must say that PSO has abundant application for mining associate rule [4,45,89].

Classification

Data clustering, one of the most important techniques in data mining, aims to group unlabeled data into different groups

on the basis of similarities and dissimilarities between the data elements. A typical clustering process involves feature selection, selection of a similarity measure, grouping of data, and assessment of the output. Alam had reviewed the research on particle swarm optimization based clustering [3]. PSO is often used in this area to optimize the parameters of traditional algorithm, like support vector machines (SVMs), backpropagation (BP) network and others. Porwik focused on signatures verification based on probabilistic neural network (PNN) classifier optimised by PSO algorithm [60]. Optimal parameters of the PNN have been determined by means of PSO procedure. Cervantes proposed a PSO-based method for SVM classification on skewed data sets [11]. PSO algorithm is used to evolve the artificial instances, eliminating noise instances for enhance the performance of support vector machines. Zhang focused on image segmentation using PSO and PCM with Mahalanobis distance [100]. PSO is used to optimize the initial clustering centers.

Prediction analysis for time series

Time series is an ordered sequence of observations that are evenly spaced at uniform time intervals and measured successively. Prediction of time series uses a sequence of historical values to develop a model for forecasting future values [46]. PSO was combined with other algorithms, like RBF neural networks, regression analysis. Lee used RBF neural networks with a nonlinear time-varying evolution PSO algorithm to realize the time series prediction in a practical power system [46]. Akande proposed a hybrid PSO and support vector regression model for modelling permeability prediction of hydrocarbon reservoir [1]. PSO is investigated for the optimal selection of SVR hyper-parameters for the first time in modelling the hydrocarbon reservoir. Zou combined least square support vector regression and PSO together for short term load forecasting in power system to solve the power dispatch problem [101].

Future research

Theory analysis

Particle swarm optimization, more widely, the swarm intelligence algorithms are based on the “trail and error” strategy. More research should be conducted on foundational problems of swarm intelligence. For example, search mechanism of swarm intelligence algorithms, the learning ability of swarm intelligence algorithms, the balance of exploration and exploitation, and more effective search strategy of algorithm should be studied.

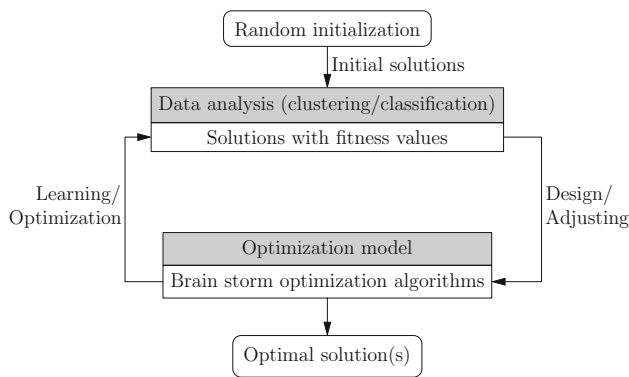


Fig. 3 A framework for data-driven swarm intelligence algorithms

Data-driven based algorithm

The data-driven algorithm indicates that the algorithm could extract the features from the solved problem and obtain the landscape by learning on the data set. Figure 3 gives a framework of data-driven swarm intelligence algorithms. Each candidate solution is a data sample from the search space. The model could be designed or adjusted via the data analysis on the previous solutions. The landscape or the difficulty of a problem could be obtained during the search, i.e., the problem could be understood better. With the learning process, more suitable algorithms could be designed to solve different problems, thus, the performance of optimization could be improved [18]. Several particle swarm optimization algorithms, especially the surrogate-assisted PSO algorithms, have been employed in data-driven algorithms, such as a surrogate-assisted PSO algorithm with committee-based active learning strategy to solve expensive problems [87], a surrogate-assisted cooperative swarm optimization to solve high-dimensional expensive problems [80].

Applications

Different optimization problems could be modeled in many areas in our everyday life. With the particle swarm optimization algorithms, or more generally swarm intelligence, more effective applications or systems can be designed to solve real-world problems. The particle swarm optimization algorithm not only could be used in problem with explicit model, but also in problem with implicit model. With the applications in complex engineering or design problems, the strength and limitation of various particle swarm optimization algorithm could be revealed and interpreted.

Conclusion

After nearly a quarter century, the particle swarm optimization algorithm has gained a great reputation and a wide range of successful applications in evolutionary computation and swarm intelligence. Particle swarm optimization algorithm, which is modeled on the social behaviors observed in flocking birds, is a population-based stochastic algorithm. In this paper, the history development, the state-of-the-art, and the applications of the PSO algorithm are reviewed. In addition, the characteristics and issues of the PSO algorithm are also discussed from the evolution and learning perspectives. Every individual in the PSO algorithm is learning from itself and another particle with good fitness value. The search performance and convergence speed are affected by different learning strategies. PSO algorithm has been widely used to solve enormous real-world problems. The scheduling and data mining problems are used as illustrations on PSO solving real-world application problems.

“Now this is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning.”² Particle swarm optimization algorithm has been invented for a quarter century, it still could be researched in many disciplines. With the analysis of different evolution and learning strategies, particle swarm optimization algorithm could be utilized on solving more real-world application problems effectively, and the strength and limitation of various PSO algorithm could be revealed.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Akande KO, Owolabi TO, Olatunji SO, AbdulRaheem A (2017) A hybrid particle swarm optimization and support vector regression model for modelling permeability prediction of hydrocarbon reservoir. *J Pet Sci Eng* 150:43–53
2. Alam S, Dobbie G, Koh YS, Riddle P (2014) Web bots detection using particle swarm optimization based clustering. In: 2014 IEEE congress on evolutionary computation (CEC), pp 2955–2962
3. Alam S, Dobbie G, Koh YS, Riddle P, Rehman SU (2014) Research on particle swarm optimization based clustering: a systematic review of literature and techniques. *Swarm Evolut Comput* 17:1–13
4. Alatas B, Akin E (2008) Rough particle swarm optimization and its applications in data mining. *Soft Comput* 12(12):1205–1218
5. Ankita S, Shikha A, Jitendra A, Sanjeev S (2013) A review on application of particle swarm optimization in association rule min-

² Winston Churchill, 10 November, 1942.

- ing. In: Satapathy SC, Udgata SK, Biswal BN (eds) Proceedings of the international conference on frontiers of intelligent computing: theory and applications (FICTA). Springer, Berlin, pp 405–414
6. Banks A, Vincent J, Anyakoha C (2008) A review of particle swarm optimization. Part II: hybridisation, combinatorial, multi-criteria and constrained optimization, and indicative applications. *Nat Comput* 7(1):109–124
 7. van den Bergh F, Engelbrecht AP (2004) A cooperative approach to particle swarm optimization. *IEEE Trans Evolut Comput* 8(3):225–239
 8. Bratton D, Kennedy J (April 2007) Defining a standard for particle swarm optimization. In: Proceedings of the 2007 IEEE swarm intelligence symposium (SIS 2007), pp 120–127
 9. Brits R, Engelbrecht AP, van den Bergh F (2003) Scalability of niche PSO. In: Proceedings of the 2003 IEEE swarm intelligence symposium (SIS 2003), pp 228–234
 10. Cai Y, Guan W, Wu Y, Xie C, Chen Y, Fang L (2017) Indoor high precision three-dimensional positioning system based on visible light communication using particle swarm optimization. *IEEE Photonics J* 9(6):1–20
 11. Cervantes J, Garcia-Lamont F, Rodriguez L, López A, Castilla JR, Trueba A (2017) PSO-based method for SVM classification on skewed data sets. *Neurocomputing* 228:187–197
 12. Chen S, Montgomery J, Bolufé-Röhler A (2015) Measuring the curse of dimensionality and its effects on particle swarm optimization and differential evolution. *Appl Intell* 42(3):514–526
 13. Chen WN, Zhang J, Lin Y, Chen N, Zhan ZH, Chung HSH, Li Y, Shi Y (2013) Particle swarm optimization with an aging leader and challengers. *IEEE Trans Evolut Comput* 17(2):241–258
 14. Chen X, Zou S (2017) Improved Wi-Fi indoor positioning based on particle swarm optimization. *IEEE Sens J* 17(21):7143–7148
 15. Cheng R, Jin Y (2015) A competitive swarm optimizer for large scale optimization. *IEEE Trans Cybern* 45(2):191–204
 16. Cheng R, Jin Y (2015) A social learning particle swarm optimization algorithm for scalable optimization. *Inf Sci* 291:43–60
 17. Cheng S, Lei X, Chen J, Feng J, Shi Y (2017) Normalized ranking based particle swarm optimizer for many objective optimization. In: Shi Y, Tan KC, Zhang M, Tang K, Li X, Zhang Q, Tan Y, Middendorf M, Jin Y (eds) Simulated evolution and learning (SEAL 2017). Springer, Berlin, pp 347–357
 18. Cheng S, Liu B, Shi Y, Jin Y, Li B (2016) Evolutionary computation and big data: key challenges and future directions. In: Tan Y, Shi Y (eds) Data mining and big data (DMBD 2016), vol 9714. Lecture notes in computer science. Springer International Publishing, Basel, pp 3–14
 19. Cheng S, Qin Q, Wu Z, Shi Y, Zhang Q (2015) Multimodal optimization using particle swarm optimization algorithms: CEC 2015 competition on single objective multi-niche optimization. In: Proceedings of the 2015 congress on evolutionary computation (CEC 2015), pp 1075–1082
 20. Cheng S, Shi Y, Qin Q (2011) Promoting diversity in particle swarm optimization to solve multimodal problems. In: Lu BL, Zhang L, Kwok J (eds) Neural information processing, (international conference on neural information processing, ICONIP 2011), vol 7063. Lecture notes in computer science. Springer, Berlin, pp 228–237
 21. Cheng S, Shi Y, Qin Q (2012) Dynamical exploitation space reduction in particle swarm optimization for solving large scale problems. In: Proceedings of 2012 IEEE congress on evolutionary computation (CEC 2012). IEEE, Brisbane, Australia, pp 3030–3037
 22. Cheng S, Shi Y, Qin Q (2012) Population diversity based study on search information propagation in particle swarm optimization. In: Proceedings of 2012 IEEE congress on evolutionary computation (CEC 2012). IEEE, Brisbane, Australia, pp 1272–1279
 23. Clerc M (2010) Beyond standard particle swarm optimisation. *Int J Swarm Intell Res (IJSIR)* 1(4):46–61
 24. Clerc M (2011) From theory to practice in particle swarm optimization. In: Panigrahi BK, Shi Y, Lim MH (eds) Handbook of swarm intelligence, adaptation, learning, and optimization, vol 8. Springer, Berlin, pp 3–36
 25. Clerc M (2012) Standard particle swarm optimisation from 2006 to 2011. Tech. rep., Independent consultant, pp 15
 26. Clerc M, Kennedy J (2002) The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans Evol Comput* 6(1):58–73
 27. Coello Coello CA, Lamont GB, Van Veldhuizen DA (2007) Evolutionary algorithms for solving multi-objective problems. In: Genetic and Evolutionary Computation Series, 2nd edn. Springer
 28. Coello Coello CA, Pulido GT, Lechuga MS (2004) Handling multiple objectives with particle swarm optimization. *IEEE Trans Evol Comput* 8(3):256–279
 29. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science, pp 39–43
 30. Eberhart R, Shi Y (2000) Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings of 2000 IEEE congress on evolutionary computation (CEC 2000). IEEE, La Jolla, CA, pp 84–88
 31. Eberhart R, Shi Y (2001) Particle swarm optimization: developments, applications and resources. In: Proceedings of the 2001 congress on evolutionary computation (CEC 2001). Seoul, pp 81–86
 32. Eberhart R, Shi Y (2007) Computational intelligence: concepts to implementations, 1st edn. Morgan Kaufmann Publishers, San Francisco
 33. Farina M, Amato P (2002) On the optimal solution definition for many-criteria optimization problems. In: Proceedings of the 2002 annual meeting of the north american fuzzy information processing society (NAFIPS-FLINT 2002), pp 233–238
 34. Feng L, Liu Z, Ma C, Huang L, Zhao L, Chen T (2008) Multi-objective vector evaluated PSO with time variant coefficients for outlier identification in power systems. In: 2008 43rd international universities power engineering conference, pp 1–6
 35. Gong YJ, Li JJ, Zhou Y, Li Y, Chung HSH, Shi Y, Zhang J (2016) Genetic learning particle swarm optimization. *IEEE Trans Cybern* 46(10):2277–2290
 36. Han H, Lu W, Qiao J (2017) An adaptive multiobjective particle swarm optimization based on multiple adaptive methods. *IEEE Trans Cybern* 47(9):2754–2767
 37. Han H, Wu X, Zhang L, Tian Y, Qiao J (2017) Self-organizing rbf neural network using an adaptive gradient multiobjective particle swarm optimization. *IEEE Trans Cybern (in press)*
 38. Ho SY, Lin HS, Liauh WH, Ho SJ (2008) OPSO: orthogonal particle swarm optimization and its application to task assignment problems. *IEEE Trans Syst Man Cybern Part A Syst Hum* 38(2):288–298
 39. Hu W, Yen GG, Luo G (2017) Many-objective particle swarm optimization using two-stage strategy and parallel cell coordinate system. *IEEE Trans Cybern* 47(6):1446–1459
 40. Hulett M, Damodaran P, Amouie M (2017) Scheduling non-identical parallel batch processing machines to minimize total weighted tardiness using particle swarm optimization. *Comput Ind Eng* 113:425–436
 41. Indira K, Kanmani S, Ashwini V, Rangalakshmi B, Divya Mary P, Sumithra M (2014) Mining association rules using adaptive particle swarm optimization. In: Mohapatra DP, Patnaik S (eds) Intelligent computing, networking, and informatics. Springer India, New Delhi, pp 975–984

42. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks, pp 1942–1948
43. Kennedy J, Eberhart R, Shi Y (2001) Swarm intelligence. Morgan Kaufmann Publishers, San Francisco
44. Kennedy J, Mendes R (2006) Neighborhood topologies in fully informed and best-of-neighborhood particle swarms. *IEEE Trans Syst Man Cybern Part C Appl Rev* 36(4):515–519
45. Kuo RJ, Chao C, Chiu Y (2011) Application of particle swarm optimization to association rule mining. *Appl Soft Comput* 11(1):326–336
46. Lee CM, Ko CN (2009) Time series prediction using RBF neural networks with a nonlinear time-varying evolution PSO algorithm. *Neurocomputing* 73(1–3):449–460
47. Lee JH, Song JY, Kim DW, Kim JW, Kim YJ, Jung SY (2017) Particle swarm optimization algorithm with intelligent particle number control for optimal design of electric machines. *IEEE Trans Ind Electron (in press)*
48. Li X, Yao X (2012) Cooperatively coevolving particle swarms for large scale optimization. *IEEE Trans Evol Comput* 16(2):210–224
49. Liang JJ, Qin AK, Suganthan PN, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans Evol Comput* 10(3):281–295
50. Liang JJ, Suganthan PN (June 2005) Dynamic multi-swarm particle swarm optimizer. In: Proceedings of the 2005 IEEE swarm intelligence symposium (SIS 2005), pp 124–129
51. Lu H, Chen X, Liu J (2012) Parallel test task scheduling with constraints based on hybrid particle swarm optimization and taboo search. *Chin J Electron* 21(4):615–618
52. Lu H, Liu J, Niu R, Zhu Z (2014) Fitness distance analysis for parallel genetic algorithm in the test task scheduling problem. *Soft Comput* 18(12):2385–2396
53. Lu H, Niu R, Liu J, Zhu Z (2013) A chaotic non-dominated sorting genetic algorithm for the multi-objective automatic test task scheduling problem. *Appl Soft Comput* 13(5):2790–2802
54. Lu H, Zhu Z, Wang X, Yin L (2014) A variable neighborhood MOEA/D for multiobjective test task scheduling problem. *Math Probl Eng* 2014:1–14
55. Maragatham G, Lakshmi M (2012) A weighted particle swarm optimization technique for optimizing association rules. In: Krishna PV, Babu MR, Ariwa E (eds) Global trends in information systems and software applications: 4th international conference, (ObCom 2011). Springer, Berlin, pp 655–664
56. Mendes R, Kennedy J, Neves J (2004) The fully informed particle swarm: simpler, maybe better. *IEEE Trans Evol Comput* 8(3):204–210
57. Mesbahi T, Rizoug N, Bartholomeüs P, Sadoun R, Khenfri F, Le Moigne P (2017) Optimal energy management for a Li-Ion battery/supercapacitor hybrid energy storage system based on a particle swarm optimization incorporating Nelder - Mead simplex approach. *IEEE Trans Intell Veh* 2(2):99–110
58. Misinem, Bakar AA, Hamdan AR, Nazri MZA (November 2010) A rough set outlier detection based on particle swarm optimization. In: 2010 10th international conference on intelligent systems design and applications, pp 1021–1025
59. Nouiri M, Bekrar A, Jemai A, Trentesaux D, Ammari AC, Niar S (2017) Two stage particle swarm optimization to solve the flexible job shop predictive scheduling problem considering possible machine breakdowns. *Comput Ind Eng* 112:595–606
60. Porwik P, Doroz R, Orczyk T (2016) Signatures verification based on PNN classifier optimised by PSO algorithm. *Pattern Recognit* 60:998–1014
61. Qin Q, Cheng S, Chu X, Lei X, Shi Y (2017) Solving nonconvex/non-smooth economic load dispatch problems via an enhanced particle swarm optimization. *Appl Soft Comput* 59:229–242
62. Qin Q, Cheng S, Zhang Q, Li L, Shi Y (2015) Biomimicry of parasitic behavior in a coevolutionary particle swarm optimization algorithm for global optimization. *Appl Soft Comput* 32:224–240
63. Qin Q, Cheng S, Zhang Q, Li L, Shi Y (2016) Particle swarm optimization with interswarm interactive learning strategy. *IEEE Trans Cybern* 46(10):2238–2251
64. Qin Q, Cheng S, Zhang Q, Wei Y, Shi Y (2015) Multiple strategies based orthogonal design particle swarm optimizer for numerical optimization. *Comput Oper Res* 60:91–110
65. Rao ACS, Somayajulu D, Banka H, Chaturvedi R (2012) Outlier detection in microarray data using hybrid evolutionary algorithm. *Proced Technol* 6:291–298
66. Ratnaweera A, Halgamuge SK, Watson HC (2004) Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Trans Evol Comput* 8(3):240–255
67. Samanta B, Nataraj C (2009) Use of particle swarm optimization for machinery fault detection. *Eng Appl Artif Intell* 22(2):308–316
68. Shahidi-Zadeh B, Tavakkoli-Moghaddam R, Taheri-Moghadam A, Rastgar I (2017) Solving a bi-objective unrelated parallel batch processing machines scheduling problem: a comparison study. *Comput Oper Res* 88:71–90
69. Shahvari O, Logendran R (2017) A bi-objective batch processing problem with dual-resources on unrelated-parallel machines. *Appl Soft Comput* 61:174–192
70. Shen L, Dauzre-Pérès S, Neufeld JS (2018) Solving the flexible job shop scheduling problem with sequence-dependent setup times. *Eur J Oper Res* 265(2):503–516
71. Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In: Proceedings of the 1998 congress on evolutionary computation, pp 69–73
72. Shi Y, Eberhart R (1999) Empirical study of particle swarm optimization. In: Proceedings of the 1999 congress on evolutionary computation, pp 1945–1950
73. Shi Y, Eberhart R (2001) Fuzzy adaptive particle swarm optimization. In: Proceedings of the 2001 congress on evolutionary computation, pp 101–106
74. Shi Y, Eberhart R (2008) Population diversity of particle swarms. In: Proceedings of the 2008 congress on evolutionary computation, pp 1063–1067
75. Shi Y, Eberhart R (2009) Monitoring of particle swarm optimization. *Front Comput Sci* 3(1):31–37
76. Shi Y, Eberhart R, Chen Y (1999) Implementation of evolutionary fuzzy system. *IEEE Trans Fuzzy Syst* 7(2):109–119
77. Shi Y (2014) Developmental swarm intelligence: Developmental learning perspective of swarm intelligence algorithms. *Int J Swarm Intell Res (IJSIR)* 5(1):36–54
78. Shi Y, Eberhart R (1998) Parameter selection in particle swarm optimization. *Evolutionary programming VII*, vol 1447. Lecture notes in computer science. Springer, Berlin, pp 591–600
79. Singh MR, Mahapatra S (2016) A quantum behaved particle swarm optimization for flexible job shop scheduling. *Comput Ind Eng* 93:36–44
80. Sun C, Jin Y, Cheng R, Ding J, Zeng J (2017) Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems. *IEEE Trans Evol Comput* 21(4):644–660
81. Ting TO, Yang XS, Cheng S, Huang K (2015) Hybrid metaheuristic algorithms: past, present, and future. In: Yang XS (ed) Recent advances in swarm intelligence and evolutionary computation, studies in computational intelligence (SCI), vol 585. Springer International Publishing, Berlin, pp 71–83
82. Tong J, Cai Y (2012) A method of aerospace test task scheduling based on genetic algorithm. In: 2012 spring congress on engineering and technology, pp 1–4

83. Tsekouras GE, Tsimikas J, Kalloniatis C, Gritzalis S (2018) Interpretability constraints for fuzzy modeling implemented by constrained particle swarm optimization. *IEEE Trans Fuzzy Syst* (**in press**)
84. Vallada E, Ruiz R (2011) A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times. *Eur J Oper Res* 211(3):612–622
85. Wachowiak MP, Timson MC, DuVal DJ (2017) Adaptive particle swarm optimization with heterogeneous multicore parallelism and GPU acceleration. *IEEE Trans Parallel Distrib Syst* 28(10):2784–2793
86. Wang H, Jiao L, Yao X (2015) Two_Arch2: an improved two-archive algorithm for many-objective optimization. *IEEE Trans Evolut Comput* 19(4):524–541
87. Wang H, Jin Y, Doherty J (2017) Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems. *IEEE Trans Cybern* 47(9):2664–2677
88. Wang H, Yang S, Ip WH, Wang D (2012) A memetic particle swarm optimisation algorithm for dynamic multi-modal optimisation problems. *Int J Syst Sci* 43(7):1268–1283
89. Wang Z, Sun X, Zhang D (2006) Classification rule mining based on particle swarm optimization. In: Wang GY, Peters JF, Skowron A, Yao Y (eds) *Rough sets and knowledge technology: first international conference (RSKT 2006)*. Springer, Berlin, pp 436–441
90. Wong LP, Puan CY, Low MYH, Chong CS (2008) Bee colony optimization algorithm with big valley landscape exploitation for job shop scheduling problems. In: *2008 winter simulation conference*, pp 2050–2058
91. Yang DL, Yang SJ (2013) Unrelated parallel-machine scheduling problems with multiple rate-modifying activities. *Inf Sci* 235:280–286
92. Ye D, Chen Z (2008) A new algorithm for high-dimensional outlier detection based on constrained particle swarm intelligence. In: Wang G, Li T, Grzymala-Busse JW, Miao D, Skowron A, Yao Y (eds) *Rough sets and knowledge technology: third international conference (RSKT 2008)*. Springer, Berlin, pp 516–523
93. Yen GG, Leong WF (2009) Dynamic multiple swarms in multiobjective particle swarm optimization. *IEEE Trans Syst Man Cybern Part A Syst Hum* 39(4):890–911
94. Yuan W, Liu Y, Wang H, Cao Y (2017) A geometric structure-based particle swarm optimization algorithm for multiobjective problems. *IEEE Trans Syst Man Cybern Syst* 47(9):2516–2537
95. Yue C, Qu B, Liang J (2018) A multi-objective particle swarm optimizer using ring topology for solving multimodal multi-objective problems. *IEEE Trans Evolut Comput* (**in press**)
96. Zhan ZH, Zhang J, Li Y, Chung HSH (2009) Adaptive particle swarm optimization. *IEEE Trans Syst Man Cybern Part B Cybern* 39(6):1362–1381
97. Zhan ZH, Zhang J, Li Y, Shi Y (2011) Orthogonal learning particle swarm optimization. *IEEE Trans Evolut Comput* 15(6):832–847
98. Zhang G, Shao X, Li P, Gao L (2009) An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem. *Comput Ind Eng* 56(4):1309–1318
99. Zhang X, Hu W, Qu W, Maybank S (2010) Multiple object tracking via species-based particle swarm optimization. *IEEE Trans Circuits Syst Video Technol* 20(11):1590–1602
100. Zhang Y, Huang D, Ji M, Xie F (2011) Image segmentation using PSO and PCM with Mahalanobis distance. *Expert Syst Appl* 38(7):9036–9040
101. Zou M, Tao H (2011) Short term load forecasting with least square support vector regression and PSO. In: Zhang J (ed) *Applied informatics and communication: international conference (ICAIC 2011)*. Springer, Berlin, pp 124–132

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.