International Journal of Network Dynamics and Intelligence Scilight

Survey/review study

A Survey of Algorithms, Applications and Trends for Particle Swarm Optimization

Jingzhong Fang [1], Weibo Liu [1,*], Linwei Chen [2], Stanislao Lauria [1], Alina Miron [1], and Xiaohui Liu [1]

[1] Department of Computer Science, Brunel University London, Uxbridge, Middlesex, UB8 3PH, United Kingdom
[2] The School of Engineering, University of Warwick, Coventry CV4 7AL, United Kingdom
[*] Correspondence: Weibo.Liu2@brunel.ac.uk

Received: 18 October 2022
Accepted: 28 November 2022
Published: 27 March 2023

Abstract: Particle swarm optimization (PSO) is a popular heuristic method, which is capable of effectively dealing with various optimization problems. A detailed overview of the original PSO and some PSO variant algorithms is presented in this paper. An up-to-date review is provided on the development of PSO variants, which include four types i.e., the adjustment of control parameters, the newly-designed updating strategies, the topological structures, and the hybridization with other optimization algorithms. A general overview of some selected applications (e.g., robotics, energy systems, power systems, and data analytics) of the PSO algorithms is also given. In this paper, some possible future research topics of the PSO algorithms are also introduced.

Keywords: particle swarm optimization; optimization; evolutionary computation; inertia weight; acceleration coefficient

1. Introduction

Optimization plays a critical role in a variety of research fields such as mathematics, economics, engineering, and computer science. Recognizing as a popular class of optimization techniques, the evolutionary computation (EC) methods have behaved competitive performance in effectively tackling optimization problems in an easy way. So far, the EC methods have been widely applied in numerous research fields thanks to their strong abilities in finding the optimal solutions [1]. Among the EC algorithms, some algorithms which are based on biological behaviours (e.g., the genetic algorithm (GA) [2], the ant colony optimization (ACO) algorithm [3] and the particle swarm optimization (PSO) algorithm [4,5]) have been well-adopted in a number of research areas e.g., energy systems, robotics, aerospace engineering and artificial intelligence.

As a population-based EC method, PSO is developed on the basis of the mimics of social behaviours e.g., the birds-flocking phenomenon and the fish-schooling phenomenon. Notably, the potential optimization solution is represented by a particle (also called as individual). During the searching process, each individual learns from the "movement experience" of itself and others. It should be mentioned that the advantages of PSO can be summarised into three aspects: (1) the number of parameters required to be adjusted is relatively few; (2) the convergence rate of the PSO algorithm is relatively fast; and (3) the implementation of the PSO algorithm is simple [6,7]. Owing to its technical merits and easy implementation, PSO has become a widely-used technique for tackling optimization problems in recent years [8−10].

Unfortunately, many population-based EC algorithms face the challenging problem that the potential solutions being easily trapped into the local optima especially under complex and high-dimensional scenarios. As a well-known EC algorithm, PSO is not an exception. As a result, developing new PSO methods has become a seemingly reasonable way to deal with the premature convergence puzzle [11−16]. For example, a group of PSO variants have been put forward by modifying the parameters [13−15]. In [14, 15], a linear decreasing mechanism has been proposed to alter the inertia weight, leading to a proper balance between the global discovery and the local detection. In [13], a novel optimizer has been introduced by presenting a time-varying strategy to adjust the acceleration coefficients, which enhances the global search and convergence performance.

Apart from modifying the control parameters, designing novel updating strategies has become a hot research direction in developing advanced PSO algorithms [17−21]. In particular, a powerful family of improved optimizers

has been proposed by embedding the switching strategy into the velocity model, thereby improving the optimal solution discovery of PSO [18,21−23]. In [21], an evolutionary factor has been designed for updating control parameters, which divides the evolution process into four different states. In [18], a switching scheme has been embedded into PSO depending on the Markov chain (which is used to determine the evolutionary states) with the aim of improving the convergence of the optimizer.

The neighbourhood information of each individual is of practical significance in finding the optimal solution. Recently, various topological structures have been designed to comprehensively utilize the neighbourhood information of each individual so as to carry out a thorough exploration in the problem space [16,24]. In [16], a variable neighbourhood operator has been introduced for improving the optimizer's search ability. A dynamically adjusted neighbourhood has been designed in [24] to enhance the information sharing in the swarm.

Hybridizing PSO with other EC algorithms is also a well-known technique in designing new PSO methods [12, 25]. For instance, in [26], the PSO algorithm has been hybridized with differential evolution to: (1) enhance the current local best particles' search strategy; and (2) enhance the possibility of individuals slipping away from the local optima. A mutation operator has been embedded into the PSO algorithm [25], which improves the convergence rate and expands the search space.

Owing to their relatively fast convergence rate and satisfactory solution quality, the PSO algorithms have been successfully applied to robot path planning, machine allocation, transportation, electricity trading, etc [27−30]. In [27], PSO has been adopted to handle the mobile robot path planning problem. A PSO-based detection approach has been put forward in [8] for finding the maximum power point in the energy power system. An improved PSO method has been developed in [30] with the purpose of handling the economic dispatch challenge in the power system by incorporating the constriction factor. A PSO-based trajectory planning approach has been introduced in [29] to find the optimal trajectory of the spacecraft. PSO has also been adopted to tackle the centroid location optimization in K-means clustering [31].

This paper aims to deliver a comprehensive and timely review of the PSO algorithms and their applications. An up-to-date review of some PSO variants is also introduced. In this paper, PSO applications in several areas are discussed. In Section 2, details of original PSO as well as basic PSO are presented. In Section 3, some existing "variant PSO algorithms" including the latest development of PSO algorithms are summarized. In Section 4, some popular practical applications of PSO algorithms are pointed out. Some possible future research topics are listed in Section 5. The conclusion is given in Section 6.

2. The PSO Algorithms

2.1. The Original PSO Algorithm

Original PSO is a prominent EC algorithm, which aims to discover the global optimum in the problem space by tuning the velocity and position of the particles [4,5]. Basically, each element of the swarm is an individual particle serving as a candidate solution. The movement of each particle is guided by (1) its previous "flying experience" which is the personal best location (i.e., pbest); and (2) the "group flying experience" which is the global best location (i.e., gbest) detected by the entire swarm.

All the individuals search the problem space which has dimensions to seek the optimal solution. The velocity and position of the i th particle at the kth iteration are denoted by $v_{i,k}$ and $x_{i,k}$, respectively. At the beginning, $v_{i,k}$ and $x_{i,k}$ are randomly initialized. During the evolution process, the velocity and position updating equations of the i th particle are given as follows:

$$\begin{aligned} v_{i,k+1} &= v_{i,k} + c_1 r_1 (pi_{i,k} - x_{i,k}) + c_2 r_2 (pg_k - x_{i,k}) \\ x_{i,k+1} &= x_{i,k} + v_{i,k+1} \end{aligned} \tag{1}$$

where k denotes the iteration number; c_1 is an acceleration factor named the cognitive parameter; c_2 is the social acceleration parameter; r_1 and r_2 are two separate random numbers selected within $[0, 1]$; $pi_{i,k}$ represents the personal best location found by the ith particle itself, which is denoted by $pi_{i,k} = [pi^1_{i,k}, pi^2_{i,k}, \cdots, pi^D_{i,k}]$; pg_k represents the global best location of all the particles, which is denoted by $pg_k = [pg^1_k, pg^2_k, \cdots, pg^D_k]$. c_1 and c_2 indicate the degree of each particle affected by itself and other particles, respectively [32]. The acceleration coefficients play an important role in balancing the local discovery and global search performance, and also have significant influence on the population diversity, solution quality, and convergence behavior of the algorithm.

2.2. The Basic PSO Algorithm

Original PSO has shown strong abilities in solving optimization problems. Nevertheless, the particles may be easily trapped in the local optimal solutions. To guarantee the search performance and balance the global detection and local discovery, the inertia weight w has been introduced in [33] as an important factor to improve the PSO algo-

rithm, and such a factor indicates the ability of particles to inherit their previous velocities. The inertia weight embedded PSO algorithm has been recognized as the basic PSO algorithm. The velocity as well as the position of the ith particle at the $(k+1)$th iteration are expressed as follows:

$$\begin{aligned} v_{i,k+1} &= wv_{i,k} + c_1 r_1 (pi_{i,k} - x_{i,k}) + c_2 r_2 (pg_k - x_{i,k}) \\ x_{i,k+1} &= x_{i,k} + v_{i,k+1} \end{aligned} \tag{2}$$

where w indicates the inertia weight. The basic PSO process is presented in Algorithm 1.

Algorithm 1 The Procedure of the Standard PSO Algorithm

1: Initialize the parameters of the PSO algorithm including the population size P, inertia weight w, acceleration coefficients c_1, c_2, and maximum velocity V_{max}

2: Set a swarm that has P particles

3: Initialize the position $x_{i,1}$ and the velocity $v_{i,1}$, and $pi_{i,1}$ of each particle ($i = 1,2,...,P$); and initialize pg_1 of the swarm

4: Calculate each particle's fitness value

5: Update the $pi_{i,k}$ of each particle and pg_k of the swarm

6: Update the velocity $v_{i,k}$ and the position $x_{i,k}$ of each particle based on Equation (2)

7: Confirm whether the maximum iterations are met or the fitness value reaches the threshold, if not, go to step 4

3. Developments of the PSO Algorithm

Similar to most population-based EC algorithms, the PSO algorithm also faces the premature convergence problem [34]. In this case, it is of practical importance to put forward new PSO algorithms especially for solving large-scale optimization problems and multi-modal optimization problems. In this paper, the reviewed PSO variants can be categorized into four groups: (1) adjusting the control parameters; (2) developing new updating strategies; (3) designing various topological structures; and (4) combining with other EC algorithms. In this section, the aforementioned four types of PSO variants are reviewed and summarized.

3.1. Adjusting Control Parameters

In the PSO algorithm, the control parameters refer to the inertia weight and the acceleration factors, which are of practical significance in maintaining the balance between global discovery and local detection. In the past few decades, plenty of work has been conducted to adjust these control parameters for improving PSO. Some selected PSO variants with modified control parameters are reviewed and summarized in Figure 1.

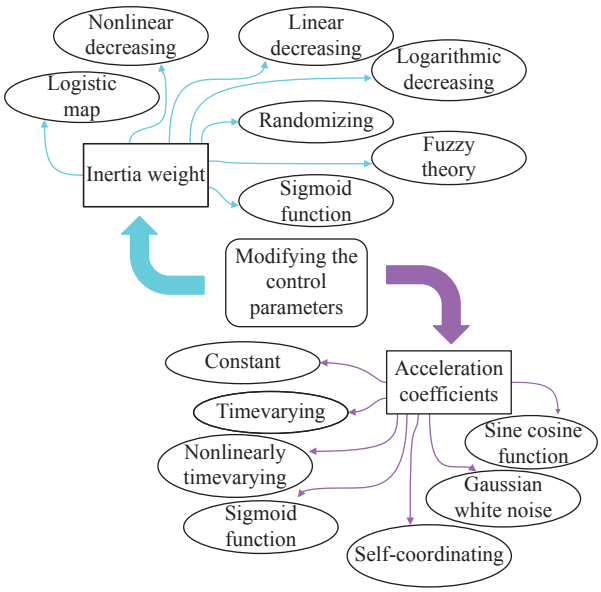

Figure 1. PSO variants with modified control parameters.

3.1.1. Inertia Weight

As an important parameter, the inertia weight is designed to achieve a proper balance between global discovery and local detection. A brief introduction is presented in Table 1 on the recently developed inertia-weight-based PSO variants.

Table 1 Inertia Weight Updating Strategies

Approach	Abbreviated Name	Reference
Linear Decreasing	Li-DIW	[14, 15]
	MIW-LD	[36]
	CLi-DIW	[44]
Non-linear Decreasing	NLFDIW	[37]
Sigmoid Function Based	SDIW	[38]
	SIIW	[39]
Logarithmic Decreasing	Lo-DIW	[40]
Randomizing	RIW	[41]
SA Algorithm Based	SAIW	[42]
Fuzzy Theory	FAPSO	[45]
Logistic Map Based	CDIW	[43]
	CRIW	[43]

It is known that a smaller inertia weight could lead to a better local search, while a larger inertia weight could contribute to a better global discovery [15]. The particles with satisfactory search ability would thoroughly exploit the solution space at the early step of search and avoid trapping into the local optima with high possibility. Additionally, the inertia weight would greatly affect the search ability of the particles. In [14, 15], starting from a relatively large value, the inertia weight is deployed to guarantee the global exploration performance. Then, the inertia weight is adjusted following the linear-decreasing strategy during the searching process with hope to enhance the local exploration. PSO with the linear decreasing inertia weight (Li-DIW) strategy has been proposed in [14, 15] where the inertia weight (w_k) is updated as follows:

$$w_k = w_{\max} - (w_{\max} - w_{\min}) \times \frac{k}{K} \tag{3}$$

where $w_{\max}$ and $w_{\min}$ denote the maximal and minimal inertia weight, respectively; k is the number of the current iteration; and K denotes the number of the maximum iteration during the evolution process. In [35], $w_{\max}$ and $w_{\min}$ are set to be 0.9 and 0.4, respectively, which becomes a normal setting in later development of PSO. The Li-DIW strategy has been widely used in developing various PSO algorithms.

It should be mentioned that a number of inertia weight updating strategies have been introduced based on the Li-DIW strategy. For example, a multi-stage inertia weight linearly decreasing strategy (MIW-LD) has been developed in [36] to better balance the global discovery and the local detection than the PSO with Li-DIW. The inertia weight (w_k) is updated by:

$$w_k = \begin{cases} \dfrac{(w_i - w_n)(k_1 - k)}{k_1} + w_n, & 0 \leqslant k < k_1 \\ w_n, & k_1 \leqslant k \leqslant k_2 \\ \dfrac{(w_n - w_f)(K - k)}{K - k_2} + w_f, & k_2 < k \leqslant K \end{cases} \tag{4}$$

where K is the maximum iteration number; k is current iteration number; w_i and w_f indicate the initial as well as the final value of the inertia weight, respectively; and w_n, k_1 and k_2 are the multi-stage parameters, which are manually selected based on experimental experience.

Different from the Li-DIW strategy, another inertia weight updating strategy is the nonlinear decreasing strategy. A nonlinear function modulated inertia weight (NLFDIW) has been developed in [37], and the updating equation of w_k is presented as follows:

$$w_k = (w_i - w_f) \times \frac{(K - k)^n}{K^n} + w_f \tag{5}$$

where n represents the nonlinear modulation index. The proposed NLFDIW could improve the convergence speed and tune the optimal solution detection strategy.

Another nonlinear function, the sigmoid function has been adopted in [38] to control the inertia weight, where a sigmoid-based decreasing inertia weight (SDIW) strategy is put forward to improve the convergence speed. The updating equation of the inertia weight (w_k) is given as:

$$w_k = \frac{w_i - w_f}{1 + \exp(-u(k - n \times K))} + w_f \tag{6}$$

where n and u are constant values to set partition of the function and to adjust the sharpness of the function, respectively.

In [39], another sigmoid function modulated inertia weight has been introduced, which is called as the sigmoid-based increasing inertia weight (SIIW). The updating equation of the inertia weight (w_k) is shown below:

$$w_k = \frac{w_i - w_f}{1 + \exp(u(k - n \times K))} + w_f \tag{7}$$

where n and u are constants to set partition of the function and to alter the function sharpness, respectively. The SIIW exhibits faster convergence rate than standard PSO.

A logarithm decreasing inertia weight (Lo-DIW) strategy has been introduced in [40] to improve the convergence rate. The updating equation of the inertia weight (w_k) is shown below:

$$w_k = w_{\max} + (w_{\min} - w_{\max}) \times \lg\left(a + \frac{10k}{K}\right) \tag{8}$$

where a is a constant value for controlling the evolutionary speed.

The global search ability becomes weak when inertia weight decreases, and the particles may fall into the local optima [6]. In the past few decades, improving the search ability of PSO has been a hot research topic by changing the inertia weight in various aspects. It is found that the PSO algorithm with Li-DIW cannot obtain satisfactory results when dealing with a nonlinear dynamic system. In this case, the random inertia weight (RIW) scheme has been proposed in [41] for tracking and optimizing a dynamic system.

Some inertia weight updating strategies are designed by using other optimization algorithms. For example, in [42], the SA integrated inertia weight (SAIW) has been introduced, where w_k is updated as follows:

$$w_k = w_{\min} + (w_{\max} - w_{\min}) \times \lambda^{k-1} \tag{9}$$

where λ is a factor used to modify the temperature parameter of the SA algorithm, which is set to be 0.95. Comparing with the standard PSO, the SAIW-based PSO shows faster convergence speed.

Using the logistic map, the chaotic-descending-based inertia weight (CDIW) strategy and the chaotic-embedded random inertia weight (CRIW) strategy have been proposed in [43]. w_k using the CDIW strategy is updated by:

$$w_k = (w_i - w_f) \times \frac{K - k}{K} + w_f \times z_k \tag{10}$$

where z_k is the logistic map. The updating equation of inertia weight (w_k) using the CRIW strategy is shown as follows:

$$w_k = 0.5 \times r \times z_k \tag{11}$$

where r is a random number selected from $[0, 1]$; and z_k is the logistic map. Both CDIW and CRIW strategies improve the convergence rate, the convergence accuracy and the global discovery ability of PSO. Comparing with the RIW, the CRIW shows better convergence rate and solution accuracy.

In [44], the chaotic-embedded LD inertia weight (CLi-DIW) strategy has been proposed, which embeds the chaotic sequences into the Li-DIW strategy. The updating equation of the inertia weight (w_k) using the CLi-DIW strategy is given by:

$$w_k = \gamma_k \times \left(w_{\max} - (w_{\max} - w_{\min}) \times \frac{k}{K}\right) \tag{12}$$

where $w_{\min}$ and $w_{\max}$ represent the minimal and maximal value of the inertia weight, respectively; k and K represent the current iteration number and the maximum iteration number, respectively; and γ_k is the chaotic parameter at the kth iteration. The CLi-DIW improves the particles' searching capability which could make it easier to slip away from the local optima.

In recent years, the fuzzy theory has been successfully applied to the PSO algorithm [45]. A fuzzy system based PSO algorithm (FAPSO) has been designed in [45], where the inputs of the fuzzy-based method are the normalized current best performance evaluation (NCBPE) as well as the current inertia weight, and the output variable of the fuzzy system is the change of the inertia weight. NCBPE is used to evaluate the best candidate solution discovered by the PSO algorithm.

3.1.2. Acceleration Coefficients

In recent years, many PSO variants which focus on the modification of the acceleration coefficients have been introduced. The reviewed acceleration coefficient updating strategies are summarized in Table 2.

Table 2 Some Acceleration Coefficient Updating Strategies

Approach	Abbreviated Name	Reference
Constant	Constant AC	[46, 47]
Time-varying	TVAC-1	[13]
	TVAC-2	[49]
	ATVAC	[50]
Non-linear Time-varying	NDAC	[51]
	NTVAC	[52]
Sigmoid Function Based	SBAC	[53]
	AWAC	[54]
Sine Cosine Function Based	SCAC	[55]
Adding Gaussian White Noises	GWNAC	[56]
Self-coordinating	SAC	[57]

The effect of two acceleration coefficients on each particle's movement is illustrated in Figure 2. According to [5], a relatively larger cognitive component could make particles search in a wider space comparing with the social component. In the original PSO algorithm, the acceleration components are constant values which are set to be 2.

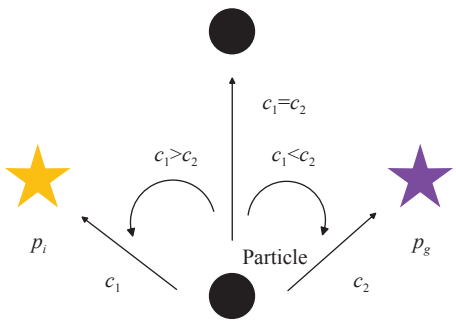

Figure 2. The effect of the value of the acceleration coefficients on the movement of the particle.

A number of new PSO methods have been put forward where different acceleration factor updating strategies have been adopted for specific optimization problems. In [46], the acceleration coefficients are set to be 2.05 to guarantee the convergence of the optimizer. In [47], c_1 and c_2 are set to be 0.5 and 2.5, respectively.

It is worth noting that a relatively larger social component may easily lead to the premature convergence problem. The time-varying acceleration coefficient-embedded scheme (TVAC-1) has been proposed in [13], where the updating equations of the acceleration factors ($c_{1,k}$ and $c_{2,k}$) are expressed as follows:

$$\begin{aligned} c_{1,k} &= c_{1i} + (c_{1f} - c_{1i}) \times \frac{k}{K} \\ c_{2,k} &= c_{2i} + (c_{2f} - c_{2i}) \times \frac{k}{K} \end{aligned} \tag{13}$$

where c_{1i} and c_{1f} indicate the initial and final values of the cognitive component, respectively; c_{2i} and c_{2f} are the initial and final values of the social component, respectively; k is the current iteration number; and K is the maximum iteration number. The time-varying scheme for altering acceleration factors could not only improve the global discovery at the early step of the searching process, but also the convergence of particles towards the globally optimal solution at the latter step of the evolution process. According to the experimental results reported in [13], the parameters are set to be $c_{1i} = 2.5$, $c_{1f} = 0.5$, $c_{2i} = 0.5$, and $c_{2f} = 2.5$. Based on the TVAC-1, a new updating strategy has been introduced to adjust the time-varying acceleration coefficients with unsymmetrical transfer range (UTRAC) in [48]. The UTRAC improves the convergence speed of the PSO algorithm.

In [49], another time-varying acceleration coefficients (TVAC-2) updating strategy has been proposed, which could further improve the solution accuracy. The updating equations of the acceleration coefficients ($c_{1,k+1}$ and $c_{2,k+1}$) using the TVAC-2 strategy are given by:

$$\begin{aligned} c_{1,k+1} &= c_{1,k} - \frac{0.5}{K} \\ c_{2,k+1} &= c_{2,k} + \frac{0.5}{K} \end{aligned} \tag{14}$$

where k and K represent the current and the maximum iteration number, respectively.

The asymmetric time-varying-based strategy for controlling acceleration coefficients (ATVAC) has been proposed in [50], where the ATVAC demonstrates merits in balancing global and local search and improving the convergence as well as the robustness. In ATVAC, the updating formulas of $c_{1,k}$ and $c_{2,k}$ are expressed by:

$$\begin{aligned} c_{1,k} &= c_{1i} - (c_{1i} - c_{1f}) \times \frac{k}{K} \\ c_{2,k} &= c_{2i} + (c_{2f} - c_{2i}) \times \frac{k}{K}. \end{aligned} \tag{15}$$

The nonlinear dynamic mechanism has been introduced in [51] to alter the acceleration factors. The updating equations of the nonlinear dynamic acceleration coefficients (NDAC) are expressed as follows:

$$\begin{aligned} c_{1,k} &= -(c_{1f} - c_{1i}) \times \left(\frac{k}{K}\right)^2 + c_{1f} \\ c_{2,k} &= c_{2i} \times \left(1 - \frac{k}{K}\right)^2 + c_{2f} \times \frac{k}{K} \end{aligned} \tag{16}$$

where c_{1i} and c_{1f} are the initial and final cognitive factors, respectively; c_{2i} and c_{2f} are the initial and final social acceleration factors, respectively; k is the current iteration number; and K is the maximum iteration number.

In [52], a set of non-linear time-varying acceleration coefficients (NTVAC) have been put forward to alleviate premature convergence. $c_{1,k}$ and $c_{2,k}$ of the PSO algorithm with NTVAC are updated by:

$$\begin{aligned} c_{1,k} &= 0.5 + 0.5 \times \exp\left(-\frac{k}{500}\right) + 1.4 \times \sin\frac{k}{30} \\ c_{2,k} &= 1 + 1.4 \times \left(1 - \exp\left(-\frac{k}{500}\right)\right) + 1.4 \times \sin\frac{k}{30} \end{aligned} \tag{17}$$

where k denotes the current iteration number.

As a popular nonlinear function, the sigmoid function has been used to adjust acceleration coefficients. In [53], sigmoid-function-based acceleration coefficients (SBAC) updating strategy has been proposed. The acceleration coefficients ($c_{1,k}$ and $c_{2,k}$) of the PSO algorithm with SBAC are updated by:

$$\begin{aligned} c_{1,k} &= \frac{1}{1+\exp\left(\frac{-\lambda k}{K}\right)} + 2(c_{1f} - c_{1i})\left(\frac{k}{K} - 1\right)^2 \\ c_{2,k} &= \frac{1}{1+\exp\left(\frac{-\lambda k}{K}\right)} + (c_{1f} - c_{1i})\left(\frac{k}{K}\right)^2 \end{aligned} \tag{18}$$

where λ is a control parameter which is set to be 0.0001.

In [54], a sigmoid-function-based adaptive weighted acceleration coefficients (AWAC) strategy has been developed, where an adaptive weighting updating function has been proposed by exploiting the distance from the particle to its pbest and gbest in order to adjust the acceleration coefficients. The updating equations of $c_{1,k}$ and $c_{2,k}$ in the PSO algorithm with AWAC are expressed as follows:

$$\begin{aligned} c_{1,k} &= \frac{b}{1+\exp(-a(l_{1,k} - c))} + d \\ c_{2,k} &= \frac{b}{1+\exp(-a(l_{2,k} - c))} + d \end{aligned} \tag{19}$$

where $l_{1,k}$ denotes the distance from an individual to its pbest at the k th iteration; $l_{2,k}$ denotes the distance between the individual and the gbest at the kth iteration; c and d are two parameters which are set to be 0 and 1.5, respectively; and a and b are two parameters used to describe the curve, which are set to be $0.000035 \times m$ and 0.5, respectively. Note that m is the range of search space of the problem.

The sine cosine acceleration coefficients (SCAC) updating strategy has been proposed in [55]. The updating equations of $c_{1,k}$ and $c_{2,k}$ in the PSO algorithm with SCAC are given by:

$$\begin{aligned} c_{1,k} &= \mu \times \sin\left(\left(1 - \frac{k}{K}\right) \times \frac{\pi}{2}\right) + \alpha \\ c_{2,k} &= \mu \times \cos\left(\left(1 - \frac{k}{K}\right) \times \frac{\pi}{2}\right) + \alpha \end{aligned} \tag{20}$$

where k and K denote the current iteration number and the maximum iteration number, respectively; and μ and α are two constants, which are set to be 2 and 0.5, respectively. SCAC could not only motivate a thorough local detection but also force the candidates moving to the global optimal solution.

Gaussian white noise-embedded acceleration coefficients (GWNAC) have been introduced in [56], where $c_{1,k}$ and $c_{2,k}$ are updated by:

$$\begin{aligned} c_{1,k} &= c_{1i} + (c_{1f} - c_{1i}) \times \frac{K-k}{K} + \delta_1 \\ c_{2,k} &= c_{2i} + (c_{2f} - c_{2i}) \times \frac{K-k}{K} + \delta_2 \end{aligned} \tag{21}$$

where δ_1 and δ_2 are two independent GWNs. By randomly perturbing the acceleration factors, the population diversity is maintained and the possibility of slipping away from the local optima is greatly enhanced.

In [57], a novel adaptive method has been proposed, where the acceleration factors are modified adaptively to make the target particle self-coordinates. The acceleration factors of the ith particle ($c_{1,i,k+1}$ and $c_{2,i,k+1}$) are updated by:

$$\begin{aligned} c_{1,i,k+1} &= c_{1,i,k} + \alpha_i (cbest_{1,k} - c_{1,i,k}), \quad i = 1, 2, \cdots, N \\ c_{2,i,k+1} &= c_{2,i,k} + \alpha_i (cbest_{2,k} - c_{2,i,k}), \quad i = 1, 2, \cdots, N \end{aligned} \tag{22}$$

where N is the swarm size; α_n represents the step size which is used to change the diversity of each particle; and $cbest_{1,k}$ and $cbest_{2,k}$ are the parameters which update the global best at certain iterations.

3.2. Developing Updating Strategies

Apart from adjusting control parameters of the PSO algorithm, many researchers have focused on designing new updating strategies of the PSO algorithm in the past few decades. The reviewed approaches on developing new algorithm updating strategies are summarized in Table 3.

Table 3 Developing New Algorithm Updating Strategies

Approach	Abbreviated Name	Reference
	VBRPSO	[59]
Re-initialization	RPSO	[17]
	RRPSO	[60]
	ERPSO	[60]
	ARPSO	[61]
	APSO	[21]
	SPSO	[18]
	ISPSO	[62]
	SDPSO	[23]
Switching Strategy	MDPSO	[63]
	ARFPSO	[64]
	DNSPSO	[24]
	RODDPSO	[22]
	FVSPSO	[66]
	SASPSO	[58]
Clustering Algorithm	CAPSO	[69]
Constriction Factor	CFPSO	[46, 70]
Comprehensive Learning	CLPSO	[34]
Multi-elitist Strategy	MEPSO	[73]
Fractional Velocity	FVPSO	[65]
FPGA Based	PMPSO	[27]
Detection Function Based	IDPSO	[74]
Quantum	QPSO	[75]
Cooperative	CPSO	[72]

3.2.1. Re-initialization

Re-initialization is a strategy which could help alleviate premature convergence [58]. In [59], a velocity-based re-initialization PSO (VBRPSO) algorithm has been presented to alleviate the premature convergence problem. In the VBRPSO algorithm, the particle velocity is monitored during the evolution process. If the median value of the norms of the velocity of the entire swarm is lower than a threshold, the swarm is treated to be stagnant and the algorithm will restart. More specifically, the stagnation can be determined by: (1) computing the Euclidean norm of the particle'

s velocity one by one; (2) sorting the obtained norms; and (3) comparing the median of the obtained norms with the pre-set threshold. If the median value of the norms is lower than the threshold, the swarm is stagnant.

In [17], an improved PSO algorithm with re-initialization mechanisms (RPSO) has been introduced, where the re-initialization process is determined based on the estimation of the varieties and activities of the particles. A new factor named "steplength" is employed to determine whether the particle should be re-initialized or not. The "steplength" for the ith particle at the kth iteration (denoted by $p_{i,k+1}$) is given by:

$$p_{i,k+1} = \sqrt{\sum_{d=1}^{N} \left(x_{i,k+1,d} - x_{i,k,d}\right)^2} \tag{23}$$

where N denotes the dimension of the problem space. If the ith particle's "steplength" is below the threshold, the ith particle will be put into an "inactive particles" group. When $P > \mu \times N$ (where P denotes the size of the "inactive particles" group, and μ is a design parameter), we need to re-initialize the particles in the "inactive particles" group. Based on a probability p_r, a number of particles are re-initialized with new velocities, positions and parameters, while other particles are re-initialized only with new parameters with hope to improve the convergence behavior and guarantee the solution accuracy.

In [60], two re-initialization methods have been proposed (which are the random re-initialization strategy and the elitist re-initialization strategy) to promote the PSO diversity. The first method is random re-initialization (RRPSO) where particles are reserved randomly, which could improve the ability of exploration. The second method is elitist re-initialization (ERPSO) where the worse preferred particles in the searching area are re-initialized, which could make the particle obtain a better fitness value than standard PSO.

3.2.2. Switching Strategy

So far, a new class of switching strategies have been designed for improving the PSO algorithm, where the evolution process is divided into several evolutionary states. In [61], an attractive and repulsive PSO (ARPSO) algorithm has been introduced, which is a diversity-guided optimizer that could alleviate the premature convergence. The velocity of the ith particle at the $(k+1)$th iteration of the ARPSO algorithm is given as below:

$$v_{i,k+1} = wv_{i,k} + a\left(r_1\left(pi_{i,k} - x_{i,k}\right) + r_2\left(pg_k - x_{i,k}\right)\right) \tag{24}$$

where w indicates the inertia weight; r_1 and r_2 are two separate numbers randomly generated within $[0,2]$; $pi_{i,k}$ represents the personal best location discovered by the ith particle itself; pg_k represents the global best location among all candidates; and a is a parameter used to determine the evolution phase. The optimizer is at the repulsion phase (which demonstrates better divergence than the other evolution phase) when a is set to be $a = -1$. The optimizer at the attraction phase would encourage the swarm to converge when a is set to be $a = 1$.

In [21], an adaptive PSO (APSO) algorithm has been proposed, where an evolutionary factor (EF) is introduced to determine the exploration, exploitation, convergence, and jumping out states. The evolutionary state estimation technique and the elitist learning strategy have been proposed in the APSO. According to the ESE, the evolution process can be divided into four states based on the evolutionary factor. The EF at the kth iteration (denoted by $E_{f,k}$) is calculated by:

$$E_{f,k} = \frac{d_{g,k} - d_{\min}}{d_{\max} - d_{\min}} \in [0,1] \tag{25}$$

where $d_{g,k}$ is the mean distance value of the global best particle at the kth iteration to all the other particles; and $d_{\min}$ and $d_{\max}$ are the minimum and the maximum value of $d_{i,k}$, which is the mean distance value of the particle i to all the other particles. Note that $d_{i,k}$ is measured by using the following equation:

$$d_{i,k} = \frac{1}{N-1} \sum_{j=1, j\neq i}^{N} \sqrt{\sum_{l=1}^{D} \left(x_{i,k,l} - x_{j,k,l}\right)^2} \tag{26}$$

where N denotes the swarm size; and D is the number of dimensions. The inertia weight (w_k) is adjusted using a sigmoid mapping based on the EF, which is shown as follows:

$$w_k = \frac{1}{1 + 1.5 \times \exp\left(-2.6E_{f,k}\right)} \in [0.4, 0.9], \quad \forall E_{f,k} \in [0,1] \tag{27}$$

where w_k is initialized to be 0.9, and the acceleration coefficients are initialized as 2 and adjusted according to the evolutionary state. The main steps of the APSO algorithm are presented in Algorithm 2. Compared with the standard PSO algorithm, the APSO algorithm demonstrates better performance in terms of the convergence speed, global optimization ability and solution accuracy.

Algorithm 2 The Procedure of the Adaptive PSO Algorithm

1: Initialize the parameters, e.g., the population size P, inertia weight $w = 0.9$, acceleration factors $c_1 = c_2 = 2$, and maximum velocity $V_{\max}$
2: Set a swarm that has P particles
3: Initialize the position $x_{i,1}$ and the velocity $v_{i,1}$, and $pi_{i,1}$ of each particle $(i = 1, 2, \ldots, P)$; and initialize pg_1 of the swarm
4: Estimate the evolutionary states and adjust the control parameters
5: Update the velocity $v_{i,k}$ and the position $x_{i,k}$ of each particle
6: Calculate the fitness of the ith particle
7: Update the $pi_{i,k}$ of each particle and pg_k of the swarm
8: Confirm whether the maximum iterations are met or the fitness value reaches the threshold, if not, go to step 4

In [18], a switching-motivated PSO algorithm (SPSO) has been developed. Based on the evolutionary factor, the velocity updating process could jump from one state to another by using a Markov chain, and the acceleration factors are thus updated based on the evolutionary state. Furthermore, a leader competitive penalized multi-learning approach is introduced in order to help the globally best particle slip away from the local optimal areas and accelerate the convergence speed. The velocity updating equation of the ith particle at the $(k+1)$th iteration of the SPSO algorithm is expressed as follows:

$$v_{i,k+1} = wv_{i,k} + c_{1,\xi_k} r_1 \left(pi_{i,k} - x_{i,k}\right) + c_{2,\xi_k} r_2 \left(pg_k - x_{i,k}\right) \tag{28}$$

where w indicates the inertia weight; c_{1,ξ_k} and c_{2,ξ_k} are acceleration coefficients; r_1 and r_2 are two separate random numbers generated within $[0, 1]$; $pi_{i,k}$ represents the personal best position found by the ith particle itself; pg_k represents the global best position of the entire swarm; and ξ_k is a non-homogeneous Markov chain which is used to determine the parameters. Different from the APSO, the mean distance value of the ith particle to all the other particles (denoted by $d_{i,k}$) is calculated as follows:

$$d_{i,k} = \frac{1}{N} \sum_{j=1}^{N} \sqrt{\sum_{l=1}^{D} \left(x_{i,k,l} - x_{j,k,l}\right)^2} \tag{29}$$

where N is the swarm size; and D is the number of dimensions. The EF can be calculated based on Equation (25). After calculating the EF, the evolutionary state can be confirmed based on the Markov chain. The parameters of the SPSO are mode-dependent. The inertia weight (w_k) of the SPSO algorithm is expressed by:

$$w_k = 0.5E_{f,k} + 0.4 \in [0.4, 0.9], \quad \forall E_{f,k} \in [0, 1] \tag{30}$$

where w_k is initialized to be 0.9; and $E_{f,k}$ is the EF. The values of the acceleration coefficients are listed in Table 4.

Table 4 Acceleration Coefficient Updating Strategies of the SPSO Algorithm

Evolutionary State	$E_{f,k}$	ξ_k	$c_{1,\xi k}$	$c_{2,\xi k}$
Convergence	[0, 0.25)	1	2	2
Exploitation	[0.25, 0.5)	2	2.1	1.9
Exploration	[0.5, 0.75)	3	2.2	1.8
Jumping-out	[0.75, 1]	4	1.8	2.2

The steps of the SPSO algorithm are presented in Algorithm 3.

Algorithm 3 The Procedure of the Switching PSO Algorithm

1: Initialize the parameters
2: Set a swarm that has P particles
3: Initialize the position $x_{i,1}$ and the velocity $v_{i,1}$, and $pi_{i,1}$ of each particle $(i = 1, 2, \ldots, P)$; and initialize pg_1 of the swarm
4: Calculate current evolutionary states based on Equation (25) and Equation (29)
5: Predict the state in the next generation
6: Adjust w according to Equation (30) and adjust c_1 and c_2 according to Table 4
7: Update the velocity $v_{i,k}$ and the position $x_{i,k}$ of each particle according to Equation (28) with Markovian jumping parameters
8: Update the $pi_{i,k}$ of each particle and pg_k of the swarm
9: Confirm whether the maximum iterations are met or the fitness value reaches the threshold, if not, go to step 4

In [24], A dynamic-neighbourhood-based SPSO (DNSPSO) algorithm has been developed, where the evolution information of the swarm is used in the velocity updating process based on (1) a distance-based dynamic neighbourhood; and (2) the switching strategy. The velocity of ith particle at the $(k+1)$th iteration can be calculated by:

$$v_{i,k+1} = wv_{i,k} + c_{1,\xi_k} r_1 \left(pi_{i,\xi_k} - x_{i,k}\right) + c_{2,\xi_k} r_2 \left(pg_{\xi_k} - x_{i,k}\right) \tag{31}$$

where w denotes the inertia weight; r_1 and r_2 are two random numbers selected from $[0,1]$; c_{1,ξ_k} and c_{2,ξ_k} are acceleration coefficients; pi_{i,ξ_k} represents the personal best position found by the ith particle itself; pg_{ξ_k} represents the global best position of the entire swarm; ξ_k denotes four different evolutionary states. Note that pi_{i,ξ_k} and pg_{ξ_k} are updated based on the dynamic neighbourhood. The evolutionary state can be calculated according to Equation (25) and Equation (26) [21]. The DNSPSO algorithm improves the solution accuracy and enhances the particle's ability of slipping away from the local optima comparing with the SPSO algorithm. The technical details of the DNSPSO algorithm are summarized in Table 5.

Table 5 Velocity Updating Strategies of the DNSPSO Algorithm

Evolutionary State	$E_{f,k}$	ξ_k	c_{1,ξ_k}	c_{2,ξ_k}
Convergence	[0, 0.25)	1	2	2
Exploitation	[0.25, 0.5)	2	2.1	1.9
Exploration	[0.5, 0.75)	3	2.2	1.8
Jumping-out	[0.75, 1]	4	1.8	2.2

Based on the SPSO algorithm, an improved SPSO (ISPSO) algorithm has been introduced in [62], where a non-stationary multistage assignment penalty function is introduced. The updating strategy of velocity jumps from each mode based on the non-homogeneous Markov chain, which uses the swarm diversity as the current search information to adjust the probability matrix in order to balance the global and local search.

In [23], a switching time-delay-embedded PSO (SDPSO) algorithm has been introduced, where the time delay is employed to alter the system dynamic of the SPSO algorithm. The utilized time delays contain the historical information of the evolutionary process. The velocity and position of the ith particle at the $(k+1)$th iteration are given as follows:

$$\begin{aligned} v_{i,k+1} &= wv_{i,k} + c_{1,\xi_k} r_1 \left(pi_{i,k-\tau_{1,\xi_k}} - x_{i,k}\right) + c_{2,\xi_k} r_2 \left(pg_{k-\tau_{2,\xi_k}} - x_{i,k}\right) \\ x_{i,k+1} &= x_{i,k} + v_{i,k+1} \end{aligned} \tag{32}$$

where c_{1,ξ_k} and c_{2,ξ_k} are acceleration coefficients; r_1 and r_2 are two separate random numbers generated within $[0,1]$; τ_{1,ξ_k} and τ_{2,ξ_k} denote the delay; $pi_{i,k}$ represents the personal best location found by the ith particle itself; pg_k represents the global best location of the entire swarm; and ξ_k is a non-homogeneous Markov chain which is used to determine the parameters. The updating strategies of the inertia weight and acceleration coefficient are same as the SPSO algorithm. The convergence speed and the global optimality of the SDPSO algorithm are competitive by comparing with some existing PSO variants.

The multi-modal delayed PSO (MDPSO) method has been developed in [63], where the so-called multi-modal time-delay is embedded into the velocity model to enlarge the search space and reduce the probability of being trapped in the local optima. In the MDPSO algorithm, the velocity updating equation of the ith particle at the $(k+1)$th iteration is given by:

$$v_{i,k+1} = wv_{i,k} + c_1 r_1 \left(pi_{i,k} - x_{i,k}\right) + c_2 r_2 \left(pg_k - x_{i,k}\right) + s_{1,k} c_3 r_3 \left(pi_{i,k-\tau i_k} - x_{i,k}\right) + s_{2,k} c_4 r_4 \left(pg_{k-\tau g_k} - x_{i,k}\right) \tag{33}$$

where w denotes the inertia weight; c_i $(i = 1,2,3,4)$ are the acceleration coefficients where $c_1 = c_3$ and $c_2 = c_4$; r_i $(i = 1,2,3,4)$ are random numbers from $[0,1]$; τi_k and τg_k indicate the randomly generated time-delays within $[0,k]$ of the local and global best particles; and $s_{1,k}$ and $s_{2,k}$ are two intensity factors. The velocity updating strategy of the MDPSO algorithm is summarized in Table 6.

Table 6 Velocity Updating Strategies of the MDPSO Algorithm

Evolutionary State	$E_{f,k}$	si_k	sg_k	τi_k	τg_k
Convergence	[0, 0.25)	0	0	-	-
Exploitation	[0.25, 0.5)	$E_{f,k}$	0	$\lfloor k \times r_{\tau i} \rfloor$	-
Exploration	[0.5, 0.75)	0	$E_{f,k}$	-	$\lfloor k \times r_{\tau g} \rfloor$
Jumping-out	[0.75, 1]	$E_{f,k}$	$E_{f,k}$	$\lfloor k \times r_{\tau i} \rfloor$	$\lfloor k \times r_{\tau g} \rfloor$

In Table 6, $\lfloor\cdot\rfloor$ represents the round down function. $r_{\tau i}$ and $r_{\tau g}$ are two random numbers uniformly selected from $[0,1]$. The MDPSO algorithm is able to reduce the probability of trapping into the local optima with satisfactory convergence rate.

A novel randomly-occurring-distributed-time-delay PSO (RODDPSO) algorithm has been proposed in [22], where the distributed time-delays are employed to perturb the dynamic behaviour of the particles. The delays occur randomly, which contributes to a better search ability than the SDPSO. The velocity of the ith particle at the $(k+1)$th iteration is updated by:

$$v_{i,k+1} = wv_{i,k} + c_1 r_1 \left(pi_{i,k} - x_{i,k}\right) + c_2 r_2 \left(pg_k - x_{i,k}\right) + m_{1,\xi_k} c_3 r_3 \sum_{\tau=1}^{S} \alpha_\tau \left(pi_{i,k-\tau} - x_{i,k}\right) + m_{2,\xi_k} c_4 r_4 \sum_{\tau=1}^{S} \alpha_\tau \left(pg_{k-\tau} - x_{i,k}\right) \tag{34}$$

where S denotes the upper bound of the distributed time-delays; α_τ is a vector with S-dimension, and each member of α_τ is chosen from 0 or 1 randomly; $pi_{i,k}$ represents the personal best position found by the ith particle itself; pg_k represents the global best position of the entire swarm; m_{1,ξ_k} and m_{2,ξ_k} indicate the intensity of the distributed time-delays; and ξ_k denotes the current evolutionary state. The evolutionary state is determined by Equation (25) and Equation (26) [21]. The velocity updating strategy of the RODDPSO algorithm at each evolutionary state is summarized in Table 7.

Table 7 Velocity Updating Strategies of the RODDPSO Algorithm

Evolutionary State	$E_{f,k}$	ξ_k	m_{1,ξ_k}	m_{2,ξ_k}
Convergence	[0, 0.25)	1	0	0
Exploitation	[0.25, 0.5)	2	0.01	0
Exploration	[0.5, 0.75)	3	0	0.01
Jumping-out	[0.75, 1]	4	0.01	0.01

In [64], a modified switching PSO algorithm with adaptive random fluctuations (ARFPSO) has been introduced, where the velocity is updated based on the evolutionary states, and the adaptive random fluctuations are added to the pbest and the gbest particles. According to the simulation results reported in [64], the ARFPSO algorithm shows superior performance in terms of search the optimal solution.

In [65], a PSO algorithm with fractional velocity (FVPSO) has been developed, where the fractional-order velocity terms are added to the velocity updating equation. The FVPSO algorithm enhances the particle's ability of jumping out of the local optima. Based on the FVPSO algorithm, an adaptive fractional-order velocity SPSO (FVSPSO) algorithm has been presented in [66], where the fractional velocity is updated based on the evolutionary state. The velocity of the ith particle at the $(k+1)$th iteration of the FVSPSO algorithm is updated by:

$$\begin{aligned} v_{i,k+1} =& \alpha_k v_{i,k} + \frac{1}{2}\alpha_k(1-\alpha_k)v_{i,k-1} + \frac{1}{6}\alpha_k(1-\alpha_k)(2-\alpha_k)v_{i,k-2} \\ &+ \frac{1}{24}\alpha_k(1-\alpha_k)(2-\alpha_k)(3-\alpha_k)v_{i,k-3} + c_1 r_1 (pi_{i,k} - x_{i,k}) + c_2 r_2 (pg_k - x_{i,k}) \end{aligned} \tag{35}$$

where c_1 and c_2 are acceleration factors; r_1, r_2 are two random numbers in $[0,1]$; $pi_{i,k}$ is the personal best position found by the ith particle itself; pg_k represents the global best position of the entire swarm; and α_k denotes the fractional order of the velocity, which can be calculated by:

$$\alpha_k = 0.9 - \frac{1}{1+\exp(-E_{f,k})} \times \frac{k}{K} \tag{36}$$

where $E_{f,k}$ represents the evolutionary factor obtained by Equation (25) and Equation (26) [21]. The FVSPSO algorithm improves the search ability and could make the particles easily jump out of the local optima.

The asynchronous PSO algorithm has been brought up in [67]. Compared with the standard PSO method, the asynchronous PSO method exhibits faster convergence rate. The asynchronous updating strategy could delay the convergence of the swarm, while the synchronous updating strategy could accelerate the convergence [68]. In [58], an adaptive switching asynchronous-synchronous PSO (SASPSO) algorithm has been proposed, where the asynchronous updating strategy and the synchronous updating strategy are hybridized, which could switch from one to another according to the fitness value of the gbest.

3.2.3. Other Strategies

In [69], the clustering algorithm has been utilized in the PSO (CAPSO) algorithm to group the particles based

on their previous best positions, and the cluster centroids are replaced by particles' personal best positions and neighbours best positions. According to the experimental results reported in [69], it is found that the CAPSO algorithm (using personal best positions as the cluster centroids) performs better than algorithms using neighbours' best positions as cluster centroids.

The constriction factor and its variants have been put forward in [46, 70, 71], which improves the convergence rate of the optimizer by limiting the motion of individuals in the optimal region. The velocity and position of the PSO algorithm with constriction factors (CFPSO) are updated by:

$$\begin{aligned} v_{i,k+1} &= \chi \times (v_{i,k} + c_1 r_1 (pi_{i,k} - x_{i,k}) + c_2 r_2 (pg_k - x_{i,k})) \\ x_{i,k+1} &= x_{i,k} + v_{i,k+1} \\ \chi &= \frac{2}{\left|2 - \phi - \sqrt{\phi^2 - 4\phi}\right|}, \quad \phi = c_1 + c_2, \phi > 4 \end{aligned} \tag{37}$$

where χ is the constriction factor; $|\cdot|$ is the absolute value of "."; r_1 and r_2 are two constants randomly generated within $[0, 1]$; and ϕ is a parameter to adjust the constriction factor where $\phi = c_1 + c_2$. Normally, ϕ and χ are set to be 4.1 and 0.729, respectively.

In [72], two cooperative PSO (CPSO) algorithms have been proposed (i.e., CPSO-S_K and CPSO-H_K), where the cooperative behaviours are utilized to improve the solution accuracy.

In [34], a comprehensive learning (CL) strategy has been proposed, where the particle's velocity is updated according to the personal best locations of all other particles. The CLPSO algorithm demonstrates better performance in solving muti-modal problems compared with the standard PSO algorithm. The velocity of the ith particle at the $(k+1)$th iteration of the CLPSO algorithm is shown as follows:

$$v_{i,k+1} = wv_{i,k} + cr_{i,k}\left(pi_{f_{i,k}} - x_{i,k}\right) \tag{38}$$

where $f_{i,k}$ represents which particle's pbest should be followed by the ith particle. The learning probability P_c decides which particle should be chosen for the ith particle to learn from. More specifically, a number of random numbers are created based on the dimensions of the ith particle. If the random number is larger than P_{c_i}, the particles will learn from its own pbest at the corresponding dimension. Otherwise, the particle will learn from another one's pbest by using a tournament selection procedure at the corresponding dimension.

In [73], a multi-elitist PSO algorithm (MEPSO) has been introduced, where the multi-elitist strategy has been employed in the PSO algorithm to improve the global search, which improves the search ability of the PSO algorithm.

In [27], a field-programmable gate array (FPGA)-based parallel meta-heuristic PSO (PMPSO) algorithm has been proposed, which employs the parallel computing strategy to run three parallel PSO algorithms in the same FPGA chip. According to the experimental results reported in [27], the PMPSO algorithm shows the merit in solving some global path planning problems.

An improved PSO algorithm with a detection function (IDPSO) has been presented in [74], where the control parameters are updated based on the detection function. The control parameters (i.e., w_k, $c_{1,k}$ and $c_{2,k}$) at the kth iteration are updated by:

$$\begin{aligned} w_k &= \frac{w_i - w_f}{1 + \exp\left(\varphi_k \times \dfrac{k - K(1 + \ln\varphi_k)}{\mu}\right)} + w_f \\ c_{1,k} &= c_1 \times \varphi_k^{-1} \\ c_{2,k} &= c_2 \times \varphi_k^{-1} \end{aligned} \tag{39}$$

where K is the maximum iteration number; w_i and w_f represent the initial and final inertia weight, respectively; φ_k represents the detection function; and μ is an adjustment factor. The IDPSO algorithm improves the search ability of the particle.

In [75], the quantum PSO (QPSO) algorithm has been proposed, where the quantum theory is introduced into the PSO algorithm. A trial method for adjusting parameters has also been proposed. A new parameter tuning method has been put forward in [76] to adjust the control parameters of the QPSO algorithm, where a global reference point has been introduced to evaluate the search range of the particle. The QPSO algorithm with the new parameter selection strategy has shown better performance than the standard QPSO algorithm in terms of convergence and solution accuracy.

3.3. Improving Topological Structures

Developing new topological structures becomes a popular way to design PSO algorithms. A lot of topological

structures have been put forward to improve the performance of the PSO algorithm. The reviewed approaches on developing new topological structures of the PSO algorithm are summarized in Table 8.

Table 8 Improving Topological Strategies of the Algorithm

Approach	Abbreviated Name	Reference
Neighborhood Operator	NOPSO	[16]
Fitness-distance-ratio Based	FDR-PSO	[81]
Dynamic Multi-swarm	DMSPSO	[82]
Hierarchical Structure	HPSO	[11]
	AHPSO	[11]
Niching	NPSO-1	[85]
	SBPSO	[86]
	ANPSO	[84]
	DLIPSO	[87]
	TPSO	[83]

The neighbour of each particle can be divided into two categories (including the local best (i.e., lbest) and the global gbest (i.e., gbest)), which is depicted in Figure 3 [77]. The particle in the lbest neighbourhood is affected by its immediate neighbours' best performance. The particle in the gbest neighbourhood is attracted to the best solution found by the entire swarm.

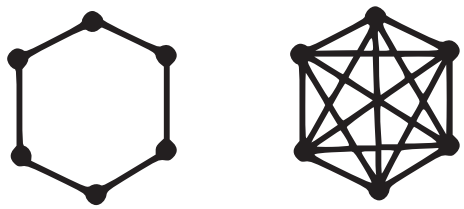

Figure 3. The lbest (left) and the gbest (right) topologies of the PSO algorithm.

The social network topology of the swarm has been modified in [78], which shows that the impact of the topology is different based on the objective function. PSO algorithms using different topological structures (including the lbest, the gbest, the pyramid, the star, the "small", and the von Neumann) have been evaluated and discussed in [79]. The experimental results reported in [79] indicate that the von Neumann configuration has consistent performance. In [80], a fully informed PSO algorithm has been put forward, where the velocity of the particle is updated according to the information from its neighbours. The network topologies of the star and the von Neumann are shown in Figure 4.

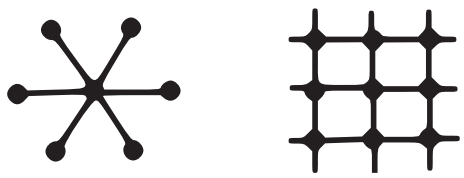

Figure 4. The star (left) and the von Neumann (right) topologies of the PSO algorithm.

In [16], the PSO algorithm with the variable neighbourhood operator (NOPSO) has been introduced, where the size of the lbest neighbourhood increases gradually until the whole swarm is connected during the evolutionary process. It should be noted that the gbest is replaced by the lbest solution to improve the search ability and avoid the local optima, which means the information is only shared locally. The neighbours can be defined by either the position compared with other particles or the proximity [32].

In [81], a fitness-distance-ratio based PSO (FDR-PSO) algorithm has been proposed, where the FDR is used to calculate the nbest. Note that nbest represents a particle's best nearest neighbour, which is used to update the particle's velocity. The velocity of the ith particle at the $(k+1)$th iteration is updated by:

$$v_{i,k+1} = wv_{i,k} + c_1(pi_{i,k} - x_{i,k}) + c_2(pg_k - x_{i,k}) + c_3(pn_k - x_{i,k}) \tag{40}$$

where w is the inertia weight; c_1, c_2 and c_3 are the acceleration coefficients; pi_d represents the personal best position found by the ith particle itself; pg_k represents the global best position of the entire swarm; and pn_k represents the historical best experience of the nbest. The velocity is updated according to three factors, which are the pbest, the gbest

and the historical best experience of the nbest. The FDR-PSO algorithm could alleviate the premature convergence during the evolutionary process.

In [82], a dynamic multi-swarm PSO (DMSPSO) algorithm has been introduced, where the swarm is divided into many small swarms and randomized frequently. Each small swarm searches the solution by itself in the search space. In order to improve the information exchange, a randomized regrouping schedule is introduced, where the population is regrouped randomly at every iteration and begins to search with a new configuration of small swarms. This neighbourhood structure shows competitiveness in solving complex multi-modal problems.

In [11], a hierarchical PSO (HPSO) algorithm has been proposed, where the particles are placed in a dynamic hierarchy and moved in the hierarchy according to the discovered best solution. The HPSO variants have been developed by adjusting the inertia weight. The updating rules of the inertia weights ($w_{1,k}$ and $w_{2,k}$) are given below:

$$w_{1,k} = \frac{(w_{\max} - w_{\min})k}{h-1} + w_{\min} \tag{41}$$

$$w_{2,k} = \frac{(w_{\min} - w_{\max})k}{h-1} + w_{\max} \tag{42}$$

where k denotes the level of the hierarchy; $w_{\max}$ and $w_{\min}$ are the maximum and minimum of the inertia weight, respectively; and h is the maximum level of the hierarchy. The $\wedge$HPSO algorithm uses Equation (41) to update the inertia weight (i.e., $w_{1,k}$), where the inertia weight of the root particle is denoted by $w_{\min}$. The inertia weight of the $\vee$HPSO algorithm (i.e., $w_{2,k}$) is updated by Equation (42), in which $w_{\max}$ represents the inertia weight of the root particle. According to the experimental results reported in [11], the HPSO algorithm shows satisfactory performance for both uni-modal and multi-modal optimization problems.

Inspired by the GA, niching is employed in PSO so as to increase the number of solutions for multi-modal optimization problems. Niching divides the swarm into several parts to explore the optimal region as much as possible. Recently, a few niching PSO methods have been developed [83–87]. In [85], a niching PSO (NPSO-1) algorithm has been proposed to solve the multi-modal problems. Each particle searches in the problem space separately until the variance of the particle's fitness values of the last three iterations is smaller than a threshold. Then, a sub-swarm of this particle and its nearest topological neighbour is generated. It is worth mentioning that other particles can join the sub-swarm when they move into the area of the sub-swarm. The NPSO-1 algorithm improves the search ability and the convergence accuracy.

In [86], a species-based PSO (SBPSO) algorithm has been developed, where all individuals are divided into a number of sub-groups of individuals based on the similarity. Different species would not share information with each other, which enhances the search performance of the algorithm when solving multi-modal optimization problems. In [84], a niching PSO algorithm with an adaptively niching parameters choosing strategy (ANPSO) has been proposed, where the statistical information of the population is utilized to adaptively update the niching parameters so as to improve the convergence rate and the solution quality of the optimizer. In [87], a distance-based locally informed PSO (DLIPSO) algorithm has been developed, where a number of lbests are utilized to guide each particle's search, which improves the search ability and avoids specifying the niching parameters.

3.4. Hybridizing with the EC Algorithms

Hybridizing with other optimization algorithms is also an important method for improving PSO algorithms. Many EC algorithms have been combined with the PSO algorithm so as to further improve the searching performance of the optimizer. The reviewed approaches on hybridizing with other EC algorithms are summarized in Table 9.

Table 9 Hybridizing the PSO Algorithm with Other EC Algorithms

Approach	Reference
Hybridizing with the GA	[25, 88, 89, 92, 93, 95–97, 98–100]
Hybridizing with the ACO Algorithm	[101]
Hybridizing the SA Algorithm	[12]
Combining with the Nelder-Mead Algorithm	[102]
Combining with the Differential Evolution	[26]

3.4.1. Hybridizing with the GA

The genetic algorithm is inspired by biological evolution, which contains lots of computational models. So far, some researchers have focused on combining the GA and the PSO algorithm to further improve the performance (e.g. convergence performance and global search ability) of the PSO algorithm.

In [88], three hybrid algorithms have been proposed, where the modification strategies using the GA are employed in the PSO algorithm. In the first strategy, the position of the gbest particle is not changed at some assigned iterations, and the crossover operation is applied to perturb the gbest. In the second strategy, the positions of pbest particles which are slow or stagnated are changed by a mutation operator. In the third strategy, the searching process is equally divided into two parts, where the first part runs the GA and the second part runs the PSO algorithm. In the PSO part, the initial swarm is assigned by using the solution of the GA. According to the experimental results reported in [88], three algorithms exhibit satisfactory convergence rate.

A PSO algorithm with the EC-based selection mechanism has been introduced in [89], where a form of tournament selection is also developed. By comparison of the particles' current fitness, the particles can be divided into two parts. The selection mechanism could change the values of the current positions and velocities of the "bad" half of the population by using the values of the "good" half of the population without changing the pbest of particles.

It is known that the mutation is an important step in the GA. Recently, the mutation operator has become an important method in developing EC algorithms, which could (1) prevent the loss of the population diversity to some extent; and (2) expand the search space [90]. The mutation operator could add new individuals to the population by creating a variation so that the population diversity can be improved [91].

In [92], a PSO algorithm with the Gaussian mutation operator has been proposed, where the dimension of the particle's position is changed by the mutation operator which obeys the Gaussian distribution. The Gaussian mutation operator (denoted by $M(x_i)$) is shown as follows:

$$M(x_i) = x_i(1+\sigma) \tag{43}$$

where σ is a random number which obeys the Gaussian distribution.

A mutation operator has been added to the PSO algorithm in [25], where a random number drawn from a Cauchy distribution is added to the component that needs to be mutated. The improved PSO algorithm with the proposed mutation operator improves the convergence rate of the optimizer and the ability of escaping from the local optima. The mutation operator (denoted by $M(x_i)$) is expressed as follows:

$$M(x_i) = x_i(1+\alpha) \tag{44}$$

where α is a random number which obeys the Cauchy distribution. The components of the particle (chosen to be mutated) is randomly selected with the probability $\frac{1}{D}$. D is the dimension of the particle.

In [93], a PSO algorithm with a nonuniform mutation operator (designed in [94]) has been developed. The operator works by changing the dimension of each individual particle. By using the nonuniform mutation operator, the performance of the PSO algorithm is improved especially for handling multi-modal problems.

In [95], a learning strategy and a mutation operator named Gaussian hyper-mutation have been added to the asynchronous PSO algorithm in order to enhance the convergence and maintain the population diversity.

The linkage is a concept in the GA algorithm. In [96], a linkage-sensitive PSO (LSPSO) algorithm has been introduced, where the elements of a linkage matrix are employed. The linkage matrix is calculated based on the performance of some randomly generated particles with perturbations. The positions of the particles which are linked are updated at the same time.

In [97], a PSO algorithm with recombination and dynamic linkage discovery (PSO-RDL) has been presented. A dynamic linkage discovery strategy is designed, where a linkage configuration is updated according to the fitness value, which is easy to implement and has a high efficiency. During the evolutionary process, a number of linkage groups are assigned, and the linkage configuration is adjusted according to the fitness value. If the average fitness value meets the threshold, the current linkage configuration will not be changed. Otherwise, the linkage groups will be reassigned randomly. In addition, a recombination operator has been designed to generate the next population by choosing and recombining building blocks from the pool randomly. The PSO-RDL algorithm shows competitive performance compared with several selected PSO variants.

In [98], a hybrid PSO algorithm with breeding and sub-populations has been developed. The population is divided into several sub-populations. Two particles are chosen randomly for breeding, and the arithmetic crossover is used during the breeding process. The parents are replaced by the offspring at the end of each iteration. The positions of the two particles (offspring) at the $(k+1)$th iteration are expressed as follows:

$$\begin{aligned} x_{i1,k+1} &= r x_{i1,k} + (1-r) x_{i2,k} \\ x_{i2,k+1} &= r x_{i2,k} + (1-r) x_{i1,k} \end{aligned} \tag{45}$$

where r is the random number in $[0,1]$. The hybrid PSO algorithm with breeding and sub-populations obtains faster convergence rate than that of the standard PSO algorithm as well as the standard GA.

A PSO algorithm with the novel multi-parent crossover operator and a self-adaptive Cauchy mutation operator

(MC-SCM-PSO) have been developed in [99], where the particle influenced by the multi-parent crossover operator can learn from three particles in the neighbourhood. The position of the offspring at the $(k+1)$th iteration of the introduced algorithm is calculated as:

$$x_{i,k+1} = r_1 x_{i,k} + r_2 x_{a,k} + r_3 x_{b,k} + r_4 x_{c,k} \tag{46}$$

where a, b, and c are three selected particles; and r_i $(i = 1,2,3,4)$ are four random numbers selected from $[-1,4]$. The MC-SCM-PSO algorithm could greatly increase the chance of slipping away from the local optima especially when solving multi-modal optimization problems.

In [100], six PSO variants with discrete crossover operators have been proposed, which choose the second parents and the number of crossover points in different ways. Experimental results show that two proposed PSO variants outperform the standard PSO algorithm.

3.4.2. Hybridizing with Other Evolutionary Methods

Apart from the GA, the PSO algorithms have also been hybridized with some other EC methods (such as the ACO algorithm, the SA algorithm and the differential evolution (DE) algorithm) with the purpose of improving its performance in convergence and solution accuracy.

In [101], a hybrid optimization algorithm which combines the FAPSO algorithm with the ACO algorithm (FAPSO-ACO) has been presented, where the control parameters of the FAPSO algorithm are adjusted according to the fuzzy rules. The decision-making structure is added to the FAPSO algorithm, which improves the performance of the PSO algorithm. In [102], the FAPSO algorithm has been combined with the Nelder-Mead (NM) simplex search, where the NM algorithm is considered as a local search algorithm to search around the global solution, which significantly improves the performance of the FAPSO algorithm.

The hybridization of PSO and SA (PSO-SA) has been introduced in [12]. The SA algorithm is utilized to search the global solution, and a mutation operator is used to enhance the communication between particles. According to the experimental results reported in [12], the PSO-SA algorithm has fast convergence rate and high accuracy.

Based on [18] and [62], the switching-local-evolutionary-based PSO (SLEPSO) algorithm has been developed in [26]. In SLEPSO, ISPSO is integrated with DE, and thus improves (1) the search ability of the current local best particles; and (2) the chance of slipping away from the local optima.

4. Applications of the PSO Algorithm

In this section, some selected practical applications of the PSO algorithm are reviewed, which are divided into 6 categories including robotics, the renewable energy system, the power system, data analytics, image processing and some other applications.

4.1. Robotics

4.1.1. Path Planning for Robots

Autonomous navigation of the mobile robot is a crucial task in robotics. The autonomous navigation process is illustrated in Figure 5. As an important task in autonomous navigation, path planning has been widely investigated, which is known as an optimization problem in certain indices with some certain constraints [103−105]. So far, a number of PSO algorithms have been adopted to solve the robots path planning problems.

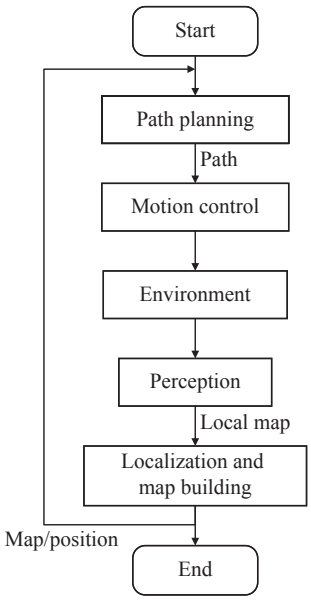

Figure 5. The autonomous navigation process of the mobile robot.

In [27], the PMPSO algorithm has been applied to deal with the navigation of the autonomous robot in structured environments with obstacles. In [106], the SLEPSO algorithm has been used to solve the path planning problem of the intelligent robot. In [63], the MDPSO algorithm has been successfully applied to the path planning for mobile robots. In [107], the novel Chaotic PSO method has been put forward to tackle the path planning problem by optimizing the control points of Bezier curve. In [64], a switching PSO algorithm with adaptive random fluctuations has been combined with the η^3-splines to solve a double-layer smooth global path planning problem with several kinematic constraints. In [66], the FVSPSO algorithm has been applied for smooth path planning for the mobile robots based on the continuous high-degree Bezier curve.

4.1.2. Robot Learning

The purpose of robot learning is to teach robots to learn skills and adapt to the environment under the guidance of learning algorithms. As an efficient class of EC algorithms, the PSO-based robot learning algorithms have been employed in the robot learning field. In [108], a PSO-based robotic learning method has been proposed, where a noise reduction technique used in the GA has been applied to the PSO algorithm to improve the performance of robot learning. In [109], an adapted PSO algorithm has been applied for unsupervised robotic learning, which shows competitiveness over some existing ones. In [110], an improved PSO-based method has been utilized in robot learning, where a statistical technique, named the optimal computing budget allocation, is utilized to improve the performance of the PSO algorithm with noises.

4.2. Renewable Energy System

In recent years, the utilization of the alternative energy has increased significantly in order to fulfil the requirements of environmental protection and electricity demands. Figure 6 shows a structure of the basic alternative energy system. In this case, the optimal designing plans of the energy system has been made to utilize the energy resources efficiently and economically. In recent years, a number of PSO-based approaches have been put forward for solving the optimization problems in energy systems [111, 112].

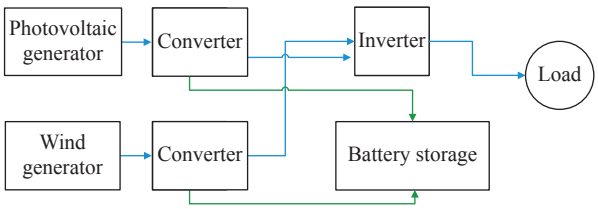

Figure 6. A basic alternative energy system with battery storage.

4.2.1. Sizing

The sizing problem is an important optimization problem in energy systems, which aims to find the optimum

number and types of devices to be used [112]. The optimum unit size can ensure the system working in the best conditions with the lowest cost, which could keep the system efficient and economic [111]. In recent years, many PSO-based methods have been developed to solve the sizing problem in energy systems.

In [113], the GA and the PSO algorithm have been used to optimize the unit size of the stand-alone hybrid energy system. A PSO-based approach has been proposed in [114] to solve the sizing problem of a hybrid solar photovoltaic energy system. In [115], the PSO algorithm has been applied to a new hybrid renewable energy system. In [116], the PSO algorithm has been applied to the sizing problem of the distributed energy system in micro grid.

4.2.2. Maximum Power Point Tracking

The maximum power point tracking (MPPT) on the solar photovoltaic module of the energy system is also a challenging optimization task, which aims to help the solar photovoltaic module produce maximum power output [112]. In recent years, a number of MPPT approaches have been developed. Among the proposed approaches, the PSO-based methods have demonstrated better noise immunity, which has become a popular way to track the maximum power point [8].

In [8], a tracking method has been proposed where the PSO algorithm has been used to recognize the maximum power point in the photovoltaic system with high efficiency and fast convergence rate. In [28], a PSO-based MPPT controller of the photovoltaic system has been introduced by using the direct control method. In [117], a MPPT method with an improved PSO algorithm has been presented where the steady-state oscillations are decreased when the maximum power point is found, which could reduce the tracking time. In [118], a 3-phase 4-wire current-controlled voltage source inverter with the PSO-based maximum power point tracking method has been developed where the PSO-based maximum power point tracking method searches independently while the control loops are working. The developed PSO-based maximum power point tracking method could improve the power quality and enhance the photovoltaic energy extraction ability. In [119], an improved PSO algorithm with the catastrophe theory has been introduced for maximum power point tracking [120]. In [121], a new maximum power point tracking approach with the PSO algorithm has been proposed, where the multiple photovoltaic arrays are controlled by only one pair of sensors. Results show that the developed method reduces the cost and improves the efficiency of the system. In [122], an MPPT algorithm has been put forward for detecting the global maximum power, where the PSO algorithm is embedded into the artificial neural network.

4.2.3. Others

In [123], a PSO algorithm with the constriction factor and the mutation operator has been applied to solve the capacity configuration problem. The PSO algorithm has been used in [124] to identify the parameters of the photovoltaic model. In [125], a parameter optimizing method of the current control strategy for a 3-phase photovoltaic grid-connected voltage source inverter system has been proposed, where the PSO algorithm is used to tuning the current control parameters. Results show that the optimized system improves the power quality.

4.3. Power System

4.3.1. Economic Dispatch

Economic dispatch (ED) problem has become a popular research topic in the power system, which plays a crucial role in the operation and planning of the power system. The ED problem aims to adjust the output of generating units in order to meet the load requirement with the lowest cost and also satisfy the operational constraints of units as well as the system [126]. In fact, the ED problem can be treated as the optimization problem, and a large number of optimization techniques have been adopted to solve the ED problem.

In [97], the PSO-RDL algorithm has been applied to the economic dispatch problem for the 3-unit power system and has discovered the currently best solution for the 40-unit system. In [127], the combined economic and emission dispatch (CEED) problems have been handled by using the PSO algorithm. In [30], an improved PSO algorithm with the constriction factor has been put forward to deal with the bid based dynamic ED problem. The PSO algorithm has been hybridized with the sequential quadratic programming (SQP) method to deal with the ED problem with the valve-point effects [128]. In [129], a CEED problem (which considers the fuel cost, the emission and the variance of generation mismatch) has been figured out by using the PSO algorithm. In [130], the PSO algorithm has been combined with the Gaussian probability distribution functions to solve the economic cost dispatch problem. In [131], an improved PSO algorithm has been employed to the economic load dispatch problem, where the inertia weight of the PSO algorithm is adaptively adjusted according to each particle's rank. The presented FAPSO-NM algorithm has been employed in [102] to figure out the economic dispatch problems of two systems consisting of 13 and 40 thermal units with satisfactory performance.

4.3.2. Others

The unit commitment (UC) is an efficient way to provide high-quality electric power to customers securely and economically [132]. The UC problem aims to find the optimal UC, which is an optimization problem. In [133], the FAPSO-based method has been developed to compute the UC of the power system.

4.4. Data Analytics

4.4.1. Clustering

Clustering algorithms organize a collection of similar items into the same cluster. An illustration of clustering is in Figure 7, where each point represents an instance in the data set. It is known that clustering plays a vital role in data analysis. Unfortunately, some studies have shown that the clustering performance of the distance-based clustering algorithms is highly dependent on the initial value of the clustering centroids and may be unsatisfactory when dealing with some complex data sets [73, 134]. In the past few years, some researchers have employed optimization algorithms to choose optimized location of the initial cluster centroids to improve the performance of distance-based clustering algorithms.

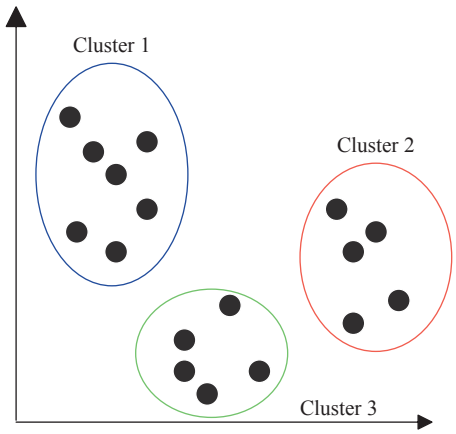

Figure 7. An illustration of a cluster analysis output.

In [31], a new K-means-based clustering method has been proposed, where the basic K-means clustering approach is combined with the Nelder-Mead simplex search approach and the standard PSO approach. In [101], the FAPSO-ACO algorithm has been embedded in improving the basic K-means approach. The RODDPSO algorithm presented in [22] has been utilized for improving the performance of the basic K-means approach. In [12], the PSO-SA algorithm has been embedded into K-means. In addition, PSO has also been utilized for optimizing the fuzzy c-means (FCM) algorithm [135, 136]. In [137], a hybrid fuzzy clustering algorithm has been proposed, where the FCM algorithm and a fuzzy PSO algorithm are combined. In [138], two improved FCM algorithms based on the IDPSO algorithm proposed in [74] have been developed, both of which demonstrate fast speed and high solution quality. A novel clustering algorithm has been introduced in [139], where the FCM algorithm is combined with the QPSO algorithm. The performance of the new method outperforms some other clustering algorithms [75].

Apart from the K-means algorithm and the FCM algorithm, some other PSO-based clustering algorithms have also been developed [140]. For example, in [141], PSO has been applied to the self-organizing maps (SOM), where a conscience factor is added to the SOM. The weights of the SOM are tuned by PSO, which could improve the robustness of the clustering algorithm. The MEPSO-based automatic kernel clustering algorithm has been introduced in [73], where the MEPSO algorithm is used to find the optimal number of clusters, and a kernel function is applied to cluster data in a high-dimensional transformed space.

4.4.2. Feature Selection

As an important member in data mining, classification aims to classify data into different categories based on the features of data. Nevertheless, selecting useful features in the data set is a difficult task. Some PSO-based approaches have been introduced with the purpose of dealing with the feature selection problems. A self-adaptive PSO algorithm has been introduced and applied in feature selection [142]. In [143], two PSO-based multi-objective feature selection methods have been proposed, where the non-dominated sorting idea is employed in the first method, and the crowding, mutation as well as dominance are applied to the second method. Both methods could automatically discover a set of non-dominated solutions.

4.5. Image Processing

Image processing is an important field in computer science, which includes image segmentation, image enhancement and so on. In recent years, a number of PSO-based algorithms have been proposed to improve the performance of image processing [9, 144−147].

4.5.1. Image Segmentation

Image segmentation focuses on analysing and interpreting the images. Many image segmentation methods have been developed by using PSO algorithms for improving the performance of image segmentation [144, 145, 147].

In [145], a multi-level thresholding method has been proposed for image segmentation, where a fractional-order Darwinian PSO algorithm is proposed to improve the accuracy of image segmentation. In [144], PSO has been combined with the hidden Markov random field model to improve the quality of the image segmentation. A PSO-based image segmentation method has been introduced in [147], where PSO is embedded into a region-based image segmentation (also named seeded region growing) method for oriented segmentation.

4.5.2. Image Enhancement

Image enhancement is another important task in image processing. The basic PSO approach has been widely adopted for image enhancement. A PSO-based automatic image enhancement technique has been proposed in [9], where PSO has been utilized to maximize an objective fitness criterion with the purpose of enhancing the contrast and details in the picture. Results show that the PSO-based automatic image enhancement technique obtains satisfactory performance in maximizing the number of pixels in the edges. A PSO-based image enhancement method has been introduced in [146], where a parameterized transformation function and an objective criterion have been utilized to improve the performance of the image enhancement.

4.6. Others

A large number of PSO-based methods have been developed for various real-world problems [148−150]. For example, the PSO algorithm has been used in [151] for parameter identification of the permanent magnet synchronous motors. In [152], the PSO algorithm has been applied in the control theory to find the optimal parameters of the proportional-integral-derivative controller in an automatic voltage regulator system, which provides fast parameter adjusting ability and is easy to implement. In [153], a FAPSO algorithm-based optimal bidding strategy of a thermal generator in the uniform price spot market has been developed to find the optimal bidding strategy in the power market, where the operating cost function and the minimum up/down constraints are considered.

PSO has been successfully employed to solve the electromagnetics-related optimization problems. In [154], an intelligence PSO algorithm has been utilized for optimizing the electromagnetic device, where the complexity of the particle is enlarged at the group level to improve the efficiency of the optimization process.

It should be pointed out that using the PSO algorithm to design the trajectory of the spacecraft has been a popular research topic. In [155], PSO has been used to handle several space trajectory optimization tasks. In [29], a PSO-based method has been proposed to find the optimal re-entry trajectories for the spacecraft. In [156], PSO has been employed to generate the end-to-end trajectory of the hypersonic re-entry vehicle. In [157], a recovery trajectory planning method has been introduced for the recovery process of the reusable launch vehicle, which combines the PSO algorithm with the polynomial guidance law.

5. Future Research Topics

Motivated by the PSO algorithms and their applications, some possible future research topics are summarized and provided:

(1) Designing new PSO algorithms for multi-objective and many-objective optimization problems. Up to now, a number of PSO algorithms have been proposed for solving multi-objective problems such as the MC-SCM-PSO algorithm and the SBPSO algorithm [86, 99]. Nevertheless, there is still room to improve the performance of the approaches. Some novel PSO algorithms of excellent performance could be extended to solve the multi-objective problems.

(2) Embedding the quantum theory into the PSO algorithm for solving continuous optimization problems. Owing to the increasing attention on the quantum theory, the quantum-based PSO algorithm has shown great research potential on enhancing the exploration capabilities. In this case, the development of the quantum-based PSO algorithm is an attractive research topic.

(3) Applying the PSO algorithms to deep learning. The study of deep learning has become a hot research direction thanks to its powerful abilities in data analytics and artificial intelligence during the past few years. The optimization of the parameters and hyper-parameters plays a critical role in affecting the performance of the deep learn-

ing systems. In this case, the deployment of PSO algorithms in deep learning systems could be a good choice.

(4) Developing new strategies and hybridizing PSO with recently proposed EC algorithms to further overcome the challenging problem of premature convergence. When introducing certain terms (e.g., time-delays and chaos) into the PSO algorithm, the investigation of the particle trajectory and the system dynamic is another research topic.

6. Conclusion

In this paper, the details of the PSO algorithms (including the original PSO algorithm and the standard PSO algorithm) have been introduced, and a comprehensive overview of some selected PSO variants has been provided. In particular, the survey of the chosen PSO variants has been presented according to four categories, namely the PSO variants with modified control parameters, the PSO variants with new updating strategies, the PSO variants with topological structures and the PSO variants hybridized with other EC algorithms. A number of real-world applications of the PSO algorithms have been reviewed and summarized. Finally, some possible research topics have been listed for future work.

Author Contributions: Jinghzong Fang: conceptualization, methodology, writing—original draft preparation and visualization; **Weibo Liu:** conceptualization, methodology, writing—original draft preparation, writing—review and editing, supervision; **Xiaohui Liu:** conceptualization and methodology; **Linwei Chen:** methodology, writing—review and editing and visualization; **Stanislao Lauria:** validation, writing—original draft preparation and writing—review and editing; **Alina Miron:** validation, writing—review and editing, visualization. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pizzuti, C. Evolutionary computation for community detection in networks: A review. *IEEE Trans. Evol. Comput.*, **2018**, *22*: 464−483.
2. Holland, J.H. *Adaptation in Natural and Artificial Systems*; University of Michigan Press: Ann Arbor, 1975.
3. Dorigo, M.; Birattari, M.; Stutzle, T. Ant colony optimization. *IEEE Comput. Intell. Mag.*, **2006**, *1*: 28−39.
4. Eberhart, R.; Kennedy, J. A new optimizer using particle swarm theory. In *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, *Nagoya*, *Japan*, *4–6 October 1995*; IEEE: Nagoya, 1995; pp. 39–43. doi: 10.1109/MHS.1995.494215
5. Kennedy, J.; Eberhart, R. Particle swarm optimization. In *Proceedings of the International Conference on Neural Networks*, *Perth*, *Australia*, *27 November 1995–1 December*; IEEE: Perth, 1995; pp. 1942–1948. doi: 10.1109/ICNN.1995.488968
6. Del Valle, Y.; Venayagamoorthy, G.K.; Mohagheghi, S.; *et al*. Particle swarm optimization: Basic concepts, variants and applications in power systems. *IEEE Trans. Evol. Comput.*, **2008**, *12*: 171−195.
7. Marini, F.; Walczak, B. Particle swarm optimization (PSO). A tutorial. *Chemom. Intell. Lab. Syst.*, **2015**, *149*: 153−165.
8. Azab, M. Optimal power point tracking for stand-alone PV system using particle swarm optimization. In *Proceedings of the 2010 IEEE International Symposium on Industrial Electronics*, *Bari*, *Italy*, *4–7 July 2010*; IEEE: Bari, 2010; pp. 969–973. doi: 10.1109/ISIE.2010.5637061
9. Braik, M.; Sheta, A.F.; Ayesh, A. Image enhancement using particle swarm optimization. In *Proceedings of the World Congress on Engineering*, *London*, *UK*, *2–4 July*, *2007*; World Congress on Engineering: London, 2007; pp. 696–701.
10. Niu, T.; Zhang, L.; Zhang, B.; *et al*. PSO-Markov residual correction method based on Verhulst-Fourier prediction model. *Syst. Sci. Control Eng*, **2021**, *9*: 32−43.
11. Janson, S.; Middendorf, M. A hierarchical particle swarm optimizer and its adaptive variant. *IEEE Trans. Syst., Man, Cybern., Part B (Cyberne.)*, **2005**, *35*: 1272−1282.
12. Niknam, T.; Amiri, B.; Olamaei, J.; *et al*. An efficient hybrid evolutionary optimization algorithm based on PSO and SA for clustering. *J. Zhejiang Univ. -Sci. A*, **2009**, *10*: 512−519.
13. Ratnaweera, A.; Halgamuge, S.K.; Watson, H.C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Trans. Evol. Comput.*, **2004**, *8*: 240−255.
14. Shi, Y.H.; Eberhart, R.C. Parameter selection in particle swarm optimization. In *Proceedings of the 7th International Conference on Evolutionary Programming*, *San Diego*, *USA*, *25–27 March 1998*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 591–600. doi: 10.1007/BFb0040810
15. Shi, Y.; Eberhart, R.C. Empirical study of particle swarm optimization. In *Proceedings of the 1999 Congress on Evolutionary Computation*, *Washington*, *USA*, *6–9 July 1999*; IEEE: Washington, 1999; pp. 1945–1950. doi: 10.1109/CEC.1999.785511
16. Suganthan, P.N. Particle swarm optimiser with neighbourhood operator. In *Proceedings of the 1999 Congress on Evolutionary Computation*, *Washington*, *USA*, *06–09 July 1999*; IEEE: Washington, 1999; pp. 1958–1962. doi: 10.1109/CEC.1999.785514
17. Guo, J.; Tang, S.J. An improved particle swarm optimization with re-initialization mechanism. In *Proceedings of the 2009 International Conference on Intelligent Human-Machine Systems and Cybernetics*, *Hangzhou*, *China*, *26–27 August 2009*; IEEE: Hangzhou, 2009; pp. 437–441. doi: 10.1109/IHMSC.2009.117
18. Tang, Y.; Wang, Z.D.; Fang, J.A. Parameters identification of unknown delayed genetic regulatory networks by a switching particle swarm optimization algorithm. *Expert Syst. Appl.*, **2011**, *38*: 2523−2535.
19. Wei, L.X.; Li, X.; Fan, R. A new multi-objective particle swarm optimisation algorithm based on R2 indicator selection mecha-

nism. *Int. J. Syst. Sci.*, **2019**, *50*: 1920−1932.
20. Xu, L.; Song, B.Y.; Cao, M.Y. An improved particle swarm optimization algorithm with adaptive weighted delay velocity. *Syst. Sci. Control Eng.*, **2021**, *9*: 188−197.
21. Zhan, Z.H.; Zhang, J.; Li, Y.; *et al*. Adaptive particle swarm optimization. *IEEE Trans. Syst., Man, Cybern., Part B (Cybern.)*, **2009**, *39*: 1362−1381.
22. Liu, W.B.; Wang, Z.D.; Liu, X.H.; *et al*. A novel particle swarm optimization approach for patient clustering from emergency departments. *IEEE Trans. Evol. Comput.*, **2019**, *23*: 632−644.
23. Zeng, N.Y.; Wang, Z.D.; Zhang, H.; *et al*. A novel switching delayed PSO algorithm for estimating unknown parameters of lateral flow immunoassay. *Cognit. Comput.*, **2016**, *8*: 143−152.
24. Zeng, N.Y.; Wang, Z.D.; Liu, W.B.; *et al*. A dynamic neighborhood-based switching particle swarm optimization algorithm. *IEEE Trans. Cybern.*, **2022**, *52*: 9290−9301.
25. Stacey, A.; Jancic, M.; Grundy, I. Particle swarm optimization with mutation. In *Proceedings of the 2003 Congress on Evolutionary Computation*, *Canberra*, *Australia*, *8 –12 December 2003*; IEEE: Canberra, 2003; pp. 1425 –1430. doi: 10.1109/CEC.2003.1299838
26. Zeng, N.Y.; Hung, Y.S.; Li, Y.R.; *et al*. A novel switching local evolutionary PSO for quantitative analysis of lateral flow immunoassay. *Expert Syst. Appl.*, **2014**, *41*: 1708−1715.
27. Huang, H.C. FPGA-based parallel metaheuristic PSO algorithm and its application to global path planning for autonomous robot navigation. *J. Intell. Robot. Syst.*, **2014**, *76*: 475−488.
28. Ishaque, K.; Salam, Z.; Shamsudin, A. Application of particle swarm optimization for maximum power point tracking of PV system with direct control method. In *Proceedings of the* 37*th Annual Conference of the IEEE Industrial Electronics Society*, *Melbourne*, *Australia*, *7–10 November 2011*; IEEE: Melbourne, 2011; pp. 1214–1219. doi: 10.1109/IECON.2011.6119482
29. Rahimi, A.; Dev Kumar, K.; Alighanbari, H. Particle swarm optimization applied to spacecraft reentry trajectory. *J. Guid., Control, Dyn.*, **2013**, *36*: 307−310.
30. Zhao, B.; Cao, Y.J. Multiple objective particle swarm optimization technique for economic load dispatch. *J. Zhejiang Univ.-Sci. A*, **2005**, *6*: 420−427.
31. Kao, Y.T.; Zahara, E.; Kao, I.W. A hybridized approach to data clustering. *Expert Syst. Appl.*, **2008**, *34*: 1754−1762.
32. Kennedy, J. The particle swarm: Social adaptation of knowledge. In *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation*, *Indianapolis*, *USA*, *13 –16 April 1997*; IEEE: Indianapolis, 1997; pp. 303 –308. doi: 10.1109/ICEC.1997.592326
33. Shi, Y.; Eberhart, R. A modified particle swarm optimizer. In *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation Proceedings*. *IEEE World Congress on Computational Intelligence*, *Anchorage*, *USA*, *4–9 May 1998*; IEEE: Anchorage, 1998; pp. 69–73. doi: 10.1109/ICEC.1998.699146
34. Liang, J.J.; Qin, A.K.; Suganthan, P.N.; *et al*. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans. Evol. Comput.*, **2006**, *10*: 281−295.
35. Bansal, J.C.; Singh, P.K.; Saraswat, M.; et al. Inertia weight strategies in particle swarm optimization. In *Proceedings of the 2011 Third World Congress on Nature and Biologically Inspired Computing*, *Salamanca*, *Spain*, *19–21 October 2011*; IEEE: Salamanca, 2011; pp. 633–640. doi: 10.1109/NaBIC.2011.6089659
36. Xin, J.B.; Chen, G.M.; Hai, Y.B. A particle swarm optimizer with multi-stage linearly-decreasing inertia weight. In *Proceedings of the 2009 International Joint Conference on Computational Sciences and Optimization*, *Sanya*, *China*, *24–26 April 2009*; IEEE: Sanya, 2009; pp. 505–508. doi: 10.1109/CSO.2009.420
37. Chatterjee, A.; Siarry, P. Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. *Comput. Oper. Res.*, **2006**, *33*: 859−871.
38. Adriansyah, A.; Amin, S.H. Analytical and empirical study of particle swarm optimization with a sigmoid decreasing inertia weight. In *Proceedings of the 2006 Postgraduate Conference on Engineering and Science*, *Johor*, *Malaysia*, *July 2006*; 2006; pp. 247 –252. https://www.researchgate.net/profile/Andi-Adriansyah/publication/277990001_Analytical_and_empirical_study_of_particle_swarm_optimization_with_a_sigmoid_decreasing_inertia_weight/links/573a7c1908aea45ee83f8fcd/Analytical-and-empirical-study-of-particle-swarm-optimization-with-a-sigmoid-decreasing-inertia-weight.pdf (accessed on 15 October 2022).
39. Malik, R.F.; Rahman, T.A.; Hashim, S.Z.M.; *et al*. New particle swarm optimizer with sigmoid increasing inertia weight. *Int. J. Comput. Sci. Secur.*, **2007**, *1*: 35−44.
40. Gao, Y.L.; An, X.H.; Liu, J.M. A particle swarm optimization algorithm with logarithm decreasing inertia weight and chaos mutation. In *Proceedings of the 2008 International Conference on Computational Intelligence and Security*, *Suzhou*, *China*, *13 –17 December 2008*; IEEE: Suzhou, 2008; pp. 61–65. doi: 10.1109/CIS.2008.183
41. Eberhart, R.C.; Shi, Y.H. Tracking and optimizing dynamic systems with particle swarms. In *Proceedings of the 2001 Congress on Evolutionary Computation*, *Seoul*, *Korea (South)*, *27 –30 May 2001*; IEEE: Seoul, 2001; pp. 94 –100. doi: 10.1109/CEC.2001.934376
42. Al-Hassan, W.; Fayek, M.B.; Shaheen, S.I. PSOSA: An optimized particle swarm technique for solving the urban planning problem. In *Proceedings of the 2006 International Conference on Computer Engineering and Systems*, *Cairo*, *Egypt*, *5–7 November 2006*; IEEE: Cairo, 2006; pp. 401–405. doi: 10.1109/ICCES.2006.320481
43. Feng, Y.; Teng, G.F.; Wang, A.X.; et al. Chaotic inertia weight in particle swarm optimization. In *Proceedings of the* 2*nd International Conference on Innovative Computing*, *Informatio and Control*, *Kumamoto*, *Japan*, *5–7 September 2007*; IEEE: Kumamoto, 2007; p. 475. doi: 10.1109/ICICIC.2007.209
44. Park, J.B.; Jeong, Y.W.; Shin, J.R.; *et al*. An improved particle swarm optimization for nonconvex economic dispatch problems. *IEEE Trans. Power Syst.*, **2010**, *25*: 156−166.
45. Shi, Y.H.; Eberhart, R.C. Fuzzy adaptive particle swarm optimization. In *Proceedings of the 2001 Congress on Evolutionary Computation*, *Seoul*, *Korea (South)*, *27–30 May 2001*; IEEE: Seoul, 2001; pp. 101–106. doi: 10.1109/CEC.2001.934377
46. Clerc, M.; Kennedy, J. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.*, **2002**, *6*: 58−73.
47. Tsai, C.Y.; Kao, I.W. Particle swarm optimization with selective particle regeneration for data clustering. *Expert Syst. Appl.*, **2011**, *38*: 6565−6576.
48. Guo, W.Z.; Chen, G.L.; Feng, X. A new strategy of acceleration coefficients for particle swarm optimization. In *Proceedings of the 10th International Conference on Computer Supported Cooperative Work in Design*, *Nanjing*, *China*, *3–5 May 2006*; IEEE: Nan-

jing, 2006; pp. 1–5. doi: 10.1109/CSCWD.2006.253100
49. Jordehi, A.R. Time varying acceleration coefficients particle swarm optimisation (TVACPSO): A new optimisation algorithm for estimating parameters of PV cells and modules. *Energy Convers. Manage.*, **2016**, *129*: 262−274.
50. Bao, G.Q.; Mao, K.F. Particle swarm optimization algorithm with asymmetric time varying acceleration coefficients. In *Proceedings of the 2009 IEEE International Conference on Robotics and Biomimetics*, *Guilin*, *China*, *19–23 December 2009*; IEEE: Guilin, 2009; pp. 2134–2139. doi: 10.1109/ROBIO.2009.5420504
51. Chen, K.; Zhou, F.Y.; Wang, Y.G.; *et al*. An ameliorated particle swarm optimizer for solving numerical optimization problems. *Appl. Soft Comput.*, **2018**, *73*: 482−496.
52. Kundu, R.; Das, S.; Mukherjee, R.; *et al*. An improved particle swarm optimizer with difference mean based perturbation. *Neurocomputing*, **2014**, *129*: 315−333.
53. Tian, D.P.; Zhao, X.F.; Shi, Z.Z. Chaotic particle swarm optimization with sigmoid-based acceleration coefficients for numerical function optimization. *Swarm Evol. Comput.*, **2019**, *51*: 100573.
54. Liu, W.B.; Wang, Z.D.; Yuan, Y.; *et al*. A novel sigmoid-function-based adaptive weighted particle swarm optimizer. *IEEE Trans. Cybern.*, **2021**, *51*: 1085−1093.
55. Chen, K.; Zhou, F.Y.; Yin, L.; *et al*. A hybrid particle swarm optimizer with sine cosine acceleration coefficients. *Inf. Sci.*, **2018**, *422*: 218−241.
56. Liu, W.B.; Wang, Z.D.; Zeng, N.Y.; *et al*. A novel randomised particle swarm optimizer. *Int. J. Mach. Learn. Cybern.*, **2021**, *12*: 529−540.
57. Yamaguchi, T.; Yasuda, K. Adaptive particle swarm optimization; self-coordinating mechanism with updating information. In *Proceedings of the 2006 IEEE International Conference on Systems*, *Man and Cybernetics*, *Taipei*, *China*, *8–11 October 2006*; IEEE: Taipei, China, 2006; pp. 2303–2308. doi: 10.1109/ICSMC.2006.385206
58. Aziz, N.A.A.; Ibrahim, Z.; Mubin, M.; *et al*. Improving particle swarm optimization via adaptive switching asynchronous – synchronous update. *Appl. Soft Comput.*, **2018**, *72*: 298−311.
59. Binkley, K.J.; Hagiwara, M. Balancing exploitation and exploration in particle swarm optimization: Velocity-based reinitialization. *Inf. Media Technol.*, **2008**, *3*: 103−111.
60. Cheng, S.; Shi, Y.H.; Qin, Q.D. Promoting diversity in particle swarm optimization to solve multimodal problems. In *Proceedings of the 18th International Conference on Neural Information Processing*, *Shanghai*, *China*, *13–17 November 2011*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 228–237. doi: 10.1007/978-3-642-24958-7_27
61. Riget, J.; Vesterstroem, J.S. A Diversity-Guided Particle Swarm Optimizer – the ARPSO; University of Aarhus: Danish, 2002. https://www.researchgate.net/publication/2532296_A_Diversity-Guided_Particle_Swarm_Optimizer_-_the_ARPSO (accessed on 15 October 2022).
62. Zeng, N.Y.; Wang, Z.D.; Li, Y.R.; *et al*. A hybrid EKF and switching PSO algorithm for joint state and parameter estimation of lateral flow immunoassay models. *IEEE/ACM Trans. Comput. Biol. Bioinf.*, **2012**, *9*: 321−329.
63. Song, B.Y.; Wang, Z.D.; Zou, L. On global smooth path planning for mobile robots using a novel multimodal delayed PSO algorithm. *Cognit. Comput.*, **2017**, *9*: 5−17.
64. Song, B.Y.; Wang, Z.D.; Zou, L.; *et al*. A new approach to smooth global path planning of mobile robots with kinematic constraints. *Int. J. Mach. Learn. Cybern.*, **2019**, *10*: 107−119.
65. Pires, E.J.S.; Machado, J.A.T.; De Moura Oliveira, P.B.; *et al*. Particle swarm optimization with fractional-order velocity. *Nonlinear Dyn.*, **2010**, *61*: 295−301.
66. Song, B.Y.; Wang, Z.D.; Zou, L. An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve. *Appl. Soft Comput.*, **2021**, *100*: 106960.
67. Carlisle, A.; Dozier, G. An off-the-shelf PSO. In *Proceedings of the Workshop on Particle Swarm Optimization*, *Indianapolis*, *USA*, *Apr*, 2001; pp. 1–6. https://www.researchgate.net/publication/216300408_An_off-the-shelf_PSO(accessed on 15 October 2022).
68. Rada-Vilela, J.; Zhang, M.J.; Seah, W. A performance study on synchronicity and neighborhood size in particle swarm optimization. *Soft Comput.*, **2013**, *17*: 1019−1030.
69. Kennedy, J. Stereotyping: Improving particle swarm performance with cluster analysis. In *Proceedings of the 2000 Congress on Evolutionary Computation*, *La Jolla*, *USA*, *16–19 July 2000*; IEEE: La Jolla, 2000; pp. 1507–1512. doi: 10.1109/CEC.2000.870832
70. Clerc, M. The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization. In *Proceedings of the 1999 Congress on Evolutionary Computation*, *Washington*, *USA*, *6–9 July 1999*; IEEE: Washington, 1999; pp. 1951–1957. doi: 10.1109/CEC.1999.785513
71. Eberhart, R.C.; Shi, Y. Comparing inertia weights and constriction factors in particle swarm optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation*, *La Jolla*, *USA*, *16–19 July 2000*; IEEE: La Jolla, 2000; pp. 84–88. doi: 10.1109/CEC.2000.870279
72. Van Den Bergh, F.; Engelbrecht, A.P. A cooperative approach to particle swarm optimization. *IEEE Trans. Evol. Comput.*, **2004**, *8*: 225−239.
73. Das, S.; Abraham, A.; Konar, A. Automatic kernel clustering with a multi-elitist particle swarm optimization algorithm. *Pattern Recognit. Lett.*, **2008**, *29*: 688−699.
74. Zhang, Y.C.; Xiong, X.; Zhang, Q.D. An improved self-adaptive PSO algorithm with detection function for multimodal function optimization problems. *Math. Probl. Eng.*, **2013**, *2013*: 716952.
75. Sun, J.; Feng, B.; Xu, W.B. Particle swarm optimization with particles having quantum behavior. In *Proceedings of the 2004 Congress on Evolutionary Computation*, *Portland*, *USA*, *19–23 June 2004*; IEEE: Portland, 2004; pp. 325–331. doi: 10.1109/CEC.2004.1330875
76. Sun, J.; Xu, W.B.; Feng, B. A global search strategy of quantum-behaved particle swarm optimization. In *Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems*, *Singapore*, *1–3 December 2004*; IEEE: Singapore, 2004; pp. 111–116. doi: 10.1109/ICCIS.2004.1460396
77. Eberhart, R.C.; Simpson, P.K.; Dobbins, R.W. *Computational Intelligence PC Tools*; Academic Press Professional: Boston, 1996.
78. Kennedy, J. Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. In *Proceedings of the 1999 Congress on Evolutionary Computation*, *Washington*, *USA*, *6–9 July 1999*; IEEE: Washington, 1999; pp. 1931–1938. doi: 10.1109/CEC.1999.785509
79. Kennedy, J.; Mendes, R. Population structure and particle swarm performance. In *Proceedings of the 2002 Congress on Evolutionary Computation*, *Honolulu*, *USA*, *12–17 May 2002*; IEEE: Honolulu, 2002; pp. 1671–1676. doi: 10.1109/CEC.2002.1004493

80. Mendes, R.; Kennedy, J.; Neves, J. The fully informed particle swarm: Simpler, maybe better. *IEEE Trans. Evol. Comput.*, **2004**, *8*: 204−210.
81. Peram, T.; Veeramachaneni, K.; Mohan, C.K. Fitness-distance-ratio based particle swarm optimization. In *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, *Indianapolis*, *USA*, *26 –26 April 2003*; IEEE: Indianapolis, 2003; pp. 174 –181. doi: 10.1109/SIS.2003.1202264
82. Liang, J.J.; Suganthan, P.N. Dynamic multi-swarm particle swarm optimizer. In *Proceedings of the 2005 IEEE Swarm Intelligence Symposium*, *Pasadena*, *USA*, *8–10 June 2005*; IEEE: Pasadena, 2005; pp. 124–129. doi: 10.1109/SIS.2005.1501611
83. Arani, B.O.; Mirzabeygi, P.; Panahi, M.S. An improved PSO algorithm with a territorial diversity-preserving scheme and enhanced exploration–exploitation balance. *Swarm Evol. Comput.*, **2013**, *11*: 1−15.
84. Bird, S.; Li, X.D. Adaptively choosing niching parameters in a PSO. In *Proceedings of the* 8*th Annual Conference on Genetic and Evolutionary Computation*, *Seattle*, *USA*, *8–12 July 2006*; ACM: Seattle, 2006; pp. 3–10. doi: 10.1145/1143997.1143999
85. Brits, R.; Engelbrecht, A.P.; Van Den Bergh, F. A niching particle swarm optimizer. In *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning*. *Orchid Country Club*, *Singapore*, *Nov*, 2002; pp. 692–696. https://www.researchgate.net/profile/Andries-Engelbrecht-2/publication/244963188_A_niching_particle_swarm_optimizer/links/542017910cf241a65a1affd8/A-niching-particle-swarm-optimizer.pdf(accessed on 15 October 2022).
86. Li, X.D. Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, *Seattle*, *USA*, *June 26 –30 2004*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 105–116. doi: 10.1007/978-3-540-24854-5_10
87. Qu, B.Y.; Suganthan, P.N.; Das, S. A distance-based locally informed particle swarm model for multimodal optimization. *IEEE Trans. Evol. Comput.*, **2013**, *17*: 387−402.
88. Premalatha, K.; Natarajan, A.M. Hybrid PSO and GA for global maximization. *Int. J. Open Problems Compt. Math.*, **2009**, *2*: 597−608.
89. Angeline, P.J. Using selection to improve particle swarm optimization. In *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation Proceedings*. *IEEE World Congress on Computational Intelligence*, *Anchorage*, *USA*, *4 –9 May 1998*; IEEE: Anchorage, 1998; pp. 84–89. doi: 10.1109/ICEC.1998.699327
90. Andrews, P.S. An investigation into mutation operators for particle swarm optimization. In *Proceedings of the 2006 IEEE International Conference on Evolutionary Computation*, *Vancouver*, *Canada*, *16–21 July 2006*; IEEE: Vancouver, 2006; pp. 1044–1051. doi: 10.1109/CEC.2006.1688424
91. Dumitrescu, D.; Lazzerini, B.; Jain, L.C.; et al. *Evolutionary Computation*; CRC Press: Boca Raton, 2000. doi: 10.1201/9781482273960
92. Higashi, N.; Iba, H. Particle swarm optimization with Gaussian mutation. In *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, *Indianapolis*, *USA*, *26–26 April 2003*; IEEE: Indianapolis, 2003; pp. 72–79. doi: 10.1109/SIS.2003.1202250
93. Esquivel, S.C.; Coello, C.A.C. On the use of particle swarm optimization with multimodal functions. In *Proceedings of the 2003 Congress on Evolutionary Computation*, *Canberra*, *Australia*, *8–12 December 2003*; IEEE: Canberra, 2003; pp. 1130–1136. doi: 10.1109/CEC.2003.1299795
94. Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs*; Springer: Berlin, 1996. doi: 10.1007/978-3-662-03315-9
95. Jiang, B.; Wang, N.; He, X.X. Asynchronous particle swarm optimizer with relearning strategy. In *Proceedings of the* 37*th Annual Conference of the IEEE Industrial Electronics Society*, *Melbourne*, *Australia*, *7–10 November 2011*; IEEE: Melbourne, 2011; pp. 2341–2346. doi: 10.1109/IECON.2011.6119675
96. Devicharan, D.; Mohan, C.K. Particle swarm optimization with adaptive linkage learning. In *Proceedings of the 2004 Congress on Evolutionary Computation*, *Portland*, *USA*, *19 –23 June 2004*; IEEE: Portland, 2004; pp. 530 –535. doi: 10.1109/CEC.2004.1330902
97. Chen, Y.P.; Peng, W.C.; Jian, M.C. Particle swarm optimization with recombination and dynamic linkage discovery. *IEEE Trans. Syst., Man, Cybern., Part B (Cybern.)*, **2007**, *37*: 1460−1470.
98. Lovbjerg, M.; Rasmussen, T.K.; Krink, T. Hybrid particle swarm optimiser with breeding and subpopulations. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, *San Francisco*, *USA*, *7–11 July 2001*; Morgan Kaufmann Publishers Inc.: San Francisco, 2001; pp. 469–476.
99. Wang, H.; Wu, Z.J.; Liu, Y.; et al. Particle swarm optimization with a novel multi-parent crossover operator. In *Proceedings of the 2008 4th International Conference on Natural Computation*, *Jinan*, *China*, *18–20 October 2008*; IEEE: Jinan, 2008; pp. 664–668. doi: 10.1109/ICNC.2008.643
100. Engelbrecht, A.P. Particle swarm optimization with discrete crossover. In *Proceedings of the 2013 IEEE Congress on Evolutionary Computation*, *Cancun*, *Mexico*, *20–23 June 2013*; IEEE: Cancun, 2013; pp. 2457–2464. doi: 10.1109/CEC.2013.6557864
101. Niknam, T.; Amiri, B. An efficient hybrid approach based on PSO, ACO and *k*-means for cluster analysis. *Appl. Soft Comput.*, **2010**, *10*: 183−197.
102. Niknam, T. A new fuzzy adaptive hybrid particle swarm optimization algorithm for non-linear, non-smooth and non-convex economic dispatch problem. *Appl. Energy*, **2010**, *87*: 327−339.
103. Atyabi, A.; Powers, D. Review of classical and heuristic-based navigation and path planning approaches. *Int. J. Adv. Comput. Technol.*, **2013**, *5*: 14.
104. Raja, P.; Pugazhenthi, S. Optimal path planning of mobile robots: A review. *Int. J. Phys. Sci.*, **2012**, *7*: 1314−1320.
105. Xu, L.; Song, B.Y.; Cao, M.Y. A new approach to optimal smooth path planning of mobile robots with continuous-curvature constraint. *Syst. Sci. Control Eng.*, **2021**, *9*: 138−149.
106. Zeng, N.Y.; Zhang, H.; Chen, Y.P.; *et al*. Path planning for intelligent robot based on switching local evolutionary PSO algorithm. *Assem. Autom.*, **2016**, *36*: 120−126.
107. Tharwat, A.; Elhoseny, M.; Hassanien, A.E.; *et al*. Intelligent Bézier curve-based path planning model using chaotic particle swarm optimization algorithm. *Cluster Comput.*, **2019**, *22*: 4745−4766.
108. Pugh, J.; Martinoli, A.; Zhang, Y. Particle swarm optimization for unsupervised robotic learning. In *Proceedings of the 2005 IEEE Swarm Intelligence Symposium*, *Pasadena*, *USA*, *8 –10 June 2005*; IEEE: Pasadena, 2005; pp. 92 –99. doi: 10.1109/SIS.2005.1501607
109. Pugh, J.; Martinoli, A. Multi-robot learning with particle swarm optimization. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*, *Hakodate*, *Japan*, *8–12 May 2006*; ACM: Hakodate, 2006; pp. 441–448. doi:

10.1145/1160633.1160715

110. Di Mario, E.; Navarro, I.; Martinoli, A. A distributed noise-resistant particle swarm optimization algorithm for high-dimensional multi-robot learning. In *2015 IEEE International Conference on Robotics and Automation*, *Seattle*, *USA*, *26–30 May 2015*; IEEE: Seattle, 2015; pp. 5970–5976. doi: 10.1109/ICRA.2015.7140036
111. Chauhan, A.; Saini, R.P. A review on integrated renewable energy system based power generation for stand-alone applications: Configurations, storage options, sizing methodologies and control. *Renewable Sustainable Energy Rev.*, **2014**, *38*: 99−120.
112. Khare, A.; Rangnekar, S. A review of particle swarm optimization and its applications in solar photovoltaic system. *Appl. Soft Comput.*, **2013**, *13*: 2997−3006.
113. Tudu, B.; Majumder, S.; Mandal, K.K.; et al. Comparative performance study of genetic algorithm and particle swarm optimization applied on off-grid renewable hybrid energy system. In *Proceedings of the 2nd International Conference on Swarm*, *Evolutionary*, *and Memetic Computing*, *Visakhapatnam*, *India*, *19–21 December 2011*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 151–158. doi: 10.1007/978-3-642-27172-4_19
114. Akshat, K.S.; Prabodh, B. Swarm intelligence based optimal sizing of solar PV, fuel cell and battery hybrid system. In *Proceedings of the 2012 International Conference on Power and Energy Systems*, *Hong Kong*, *China*, *August 2012*; Information Engineering Research Institute: Hong Kong, China, 2012; pp. 467–473.
115. Bashir, M.; Sadeh, J. Size optimization of new hybrid stand-alone renewable energy system considering a reliability index. In *Proceedings of the 11th International Conference on Environment and Electrical Engineering*, *Venice*, *Italy*, *18–25 May 2012*; IEEE: Venice, 2009; pp. 989–994. doi: 10.1109/EEEIC.2012.6221521
116. Navaeefard, A.; Tafreshi, S.M.M.; Barzegari, M.; et al. Optimal sizing of distributed energy resources in microgrid considering wind energy uncertainty with respect to reliability. In *Proceedings of the 2010 IEEE International Energy Conference*, *Manama*, *Bahrain*, *18–22 December 2010*; IEEE: Manama, 2010; pp. 820–825. doi: 10.1109/ENERGYCON.2010.5771795
117. Ishaque, K.; Salam, Z.; Amjad, M.; *et al*. An improved particle swarm optimization (PSO)–based MPPT for PV with reduced steady-state oscillation. *IEEE Trans. Power Electron.*, **2012**, *27*: 3627−3638.
118. Tumbelaka, H.H.; Miyatake, M. A grid current-controlled inverter with particle swarm optimization MPPT for PV generators. *World Acad. Sci., Eng. Technol.*, **2010**, *4*: 1086−1091.
119. Fu, Q.; Tong, N. A new PSO algorithm based on adaptive grouping for photovoltaic MPP prediction. In *Proceedings of the* 2*nd International Workshop on Intelligent Systems and Applications*, *Wuhan*, *China*, *22–23 May 2010*; IEEE: Wuhan, 2010; pp. 1–5. doi: 10.1109/IWISA.2010.5473243
120. Rosa, J.; Canovas, P.; Islam, A.; *et al*. Survivin modulates microtubule dynamics and nucleation throughout the cell cycle. *Mol. Biol. Cell*, **2006**, *17*: 1483−1493.
121. Miyatake, M.; Veerachary, M.; Toriumi, F.; *et al*. Maximum power point tracking of multiple photovoltaic arrays: A PSO approach. *IEEE Trans. Aerosp. Electron. Syst.*, **2011**, *47*: 367−380.
122. Ngan, M.S.; Tan, C.W. Multiple peaks tracking algorithm using particle swarm optimization incorporated with artificial neural network. *Int. J. Electr., Electron. Commun. Sci.*, **2011**, *5*: 1297−1303.
123. Zhao, Y.S.; Zhan, J.; Zhang, Y.; et al. The optimal capacity configuration of an independent Wind/PV hybrid power supply system based on improved PSO algorithm. In *Proceedings of the* 8*th International Conference on Advances in Power System Control*, *Operation and Management*, *Hong Kong*, *China*, *8–11 November 2009*; IEEE: Hong Kong, China, 2009; pp. 1–7. doi: 10.1049/cp.2009.1806
124. Soon, J.J.; Low, K.S. Optimizing photovoltaic model parameters for simulation. In *Proceedings of the 2012 IEEE International Symposium on Industrial Electronics*, *Hangzhou*, *China*, *28–31 May 2012*; IEEE: Hangzhou, 2012; pp. 1813–1818. doi: 10.1109/ISIE.2012.6237367
125. Al-Saedi, W.; Lachowicz, S.W.; Habibi, D. An optimal current control strategy for a three-phase grid-connected photovoltaic system using particle swarm optimization. In *Proceedings of the 2011 IEEE Power Engineering and Automation Conference*, *Wuhan*, *China*, *8–9 September 2011*; IEEE: Wuhan, 2011; pp. 286–290. doi: 10.1109/PEAM.2011.6134857
126. Mahor, A.; Prasad, V.; Rangnekar, S. Economic dispatch using particle swarm optimization: A review. *Renewable Sustainable Energy Rev.*, **2009**, *13*: 2134−2141.
127. Kumar, A.I.S.; Dhanushkodi, K.; Kumar, J.J.; et al. Particle swarm optimization solution to emission and economic dispatch problem. In *Proceedings of the 2003 Conference on Convergent Technologies for Asia-Pacific Region*, *Bangalore*, *India*, *15–17 October 2003*; IEEE: Bangalore, 2003; pp. 435–439. doi: 10.1109/TENCON.2003.1273360
128. Victoire, T.A.A.; Jeyakumar, A.E. Reserve constrained dynamic dispatch of units with valve-point effects. *IEEE Trans. Power Syst.*, **2005**, *20*: 1273−1282.
129. Umayal, S.P.; Kamaraj, N. Stochastic multi objective short term hydrothermal scheduling using particle swarm optimization. In *Proceedings of the 2005 Annual IEEE India Conference*, *Chennai*, *India*, *11–13 December 2005*; IEEE: Chennai, 2005; pp. 497–501. doi: 10.1109/INDCON.2005.1590220
130. Coelho, L.D.S.; Lee, C.S. Solving economic load dispatch problems in power systems using chaotic and Gaussian particle swarm optimization approaches. *Int. J. Elect. Power Energy Syst.*, **2008**, *30*: 297−307.
131. Panigrahi, B.K.; Pandi, V.R.; Das, S. Adaptive particle swarm optimization approach for static and dynamic economic load dispatch. *Energy Convers. Manage.*, **2008**, *49*: 1407−1415.
132. Padhy, N.P. Unit commitment-a bibliographical survey. *IEEE Trans. Power Syst.*, **2004**, *19*: 1196−1205.
133. Saber, A.Y.; Senjyu, T.; Yona, A.; *et al*. Unit commitment computation by fuzzy adaptive particle swarm optimisation. *IET Gener., Transm. Distrib.*, **2007**, *1*: 456−465.
134. Karaboga, D.; Ozturk, C. A novel clustering approach: Artificial bee colony (ABC) algorithm. *Appl. Soft Comput.*, **2011**, *11*: 652−657.
135. Bezdek, J.C. *Pattern Recognition with Fuzzy Objective Function Algorithms*; Springer Science & Business Media: New York, 1981. https://books.google.pt/books?hl=zh-CN&lr=&id=z6XqBwAAQBAJ&oi=fnd&pg=PR14&dq=Pattern+Recognition+with+Fuzzy+Objective+Function+Algorithms&ots=0i1IuVAoGr&sig=LwLEjFoZxDj9FtFMZsshIoGa9w8&redir_esc=y#v=onepage&q=Pattern%20Recognition%20with%20Fuzzy%20Objective%20Function%20Algorithms&f=false(accessed on 15 October 2022).
136. Li, L.L.; Liu, X.Y.; Xu, M.M. A novel fuzzy clustering based on particle swarm optimization. In *Proceedings of the 1st IEEE International Symposium on Information Technologies and Applications in Education*, *Kunming*, *China*, *23–25 November 2007*; IEEE: Kunming, 2007; pp. 88–90. doi: 10.1109/ISITAE.2007.4409243
137. Izakian, H.; Abraham, A.; Snášel, V. Fuzzy clustering using hybrid fuzzy c-means and fuzzy particle swarm optimization. In *Pro-*

ceedings of the 2009 World Congress on Nature & Biologically Inspired Computing, *Coimbatore*, *India*, *9–11 December 2009*; IEEE: Coimbatore, 2009; pp. 1690–1694. doi: 10.1109/NABIC.2009.5393618

138. Filho, T.M.S.; Pimentel, B.A.; Souza, R.M.C.R.; *et al*. Hybrid methods for fuzzy clustering based on fuzzy c-means and improved particle swarm optimization. *Expert Syst. Appl.*, **2015**, *42*: 6315−6328.
139. Sengupta, S.; Basak, S.; Peters, R.A. Data clustering using a hybrid of fuzzy C-means and quantum-behaved particle swarm optimization. In *Proceedings of the 2018 IEEE 8th Annual Computing and Communication Workshop and Conference*, *Las Vegas*, *USA*, *8–10 January 2018*; IEEE: Las Vegas, 2018; pp. 137–142. doi: 10.1109/CCWC.2018.8301693
140. Alam, S.; Dobbie, G.; Koh, Y.S.; *et al*. Research on particle swarm optimization based clustering: A systematic review of literature and techniques. *Swarm Evol. Comput.*, **2014**, *17*: 1−13.
141. Xiao, X.; Dow, E.R.; Eberhart, R.; et al. Gene clustering using self-organizing maps and particle swarm optimization. In *Proceedings of the International Parallel and Distributed Processing Symposium*, *Nice*, *France*, *22–26 April 2003*; IEEE: Nice, 2003; pp. 10–19. doi: 10.1109/IPDPS.2003.1213290
142. Xue, Y.; Xue, B.; Zhang, M.J. Self-adaptive particle swarm optimization for large-scale feature selection in classification. *ACM Trans. Knowl. Discovery Data*, **2019**, *13*: 50.
143. Xue, B.; Zhang, M.J.; Browne, W.N. Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE Trans. Cybern.*, **2013**, *43*: 1656−1671.
144. Ait-Aoudia, S.; Guerrout, E.H.; Mahiou, R. Medical image segmentation using particle swarm optimization. In *Proceedings of the 18th International Conference on Information Visualisation*, *Paris*, *France*, *16–18 July 2014*; IEEE: Paris, 2014; pp. 287–291. doi: 10.1109/IV.2014.68
145. Ghamisi, P.; Couceiro, M.S.; Martins, F.M.L.; *et al*. Multilevel image segmentation based on fractional-order Darwinian particle swarm optimization. *IEEE Trans. Geosci. Remote Sens.*, **2014**, *52*: 2382−2394.
146. Gorai, A.; Ghosh, A. Gray-level image enhancement by particle swarm optimization. In *Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing*, *Coimbatore*, *India*, *9–11 December 2009*; IEEE: Coimbatore, 2009; pp. 72–77. doi: 10.1109/NABIC.2009.5393603
147. Mohsen, F.; Hadhoud, M.M.; Mostafa, K.; *et al*. A new image segmentation method based on particle swarm optimization. *Int. Arab J. Inf. Technol.*, **2012**, *9*: 487−493.
148. Song, B.Y.; Xiao, Y.H.; Xu, L. Design of fuzzy PI controller for brushless DC motor based on PSO–GSA algorithm. *Syst. Sci. Control Eng.*, **2020**, *8*: 67−77.
149. Zhang, P.; Lai, X.Z.; Wang, Y.W.; *et al*. PSO-based nonlinear model predictive planning and discrete-time sliding tracking control for uncertain planar underactuated manipulators. *Int. J. Syst. Sci.*, **2022**, *53*: 2075−2089.
150. Zong, T.C.; Li, J.H.; Lu, G.P. Bias-compensated least squares and fuzzy PSO based hierarchical identification of errors-in-variables Wiener systems. *Int. J. Syst. Sci*. **2022**, in press, doi: 10.1080/00207721.2022.2135976.
151. Liu, L.; Liu, W.X.; Cartes, D.A. Particle swarm optimization-based parameter identification applied to permanent magnet synchronous motors. *Eng. Appl. Artif. Intell.*, **2008**, *21*: 1092−1100.
152. Gaing, Z.L. A particle swarm optimization approach for optimum design of PID controller in AVR system. *IEEE Trans. Energy Convers.*, **2004**, *19*: 384−391.
153. Bajpai, P.; Singh, S.N. Fuzzy adaptive particle swarm optimization for bidding strategy in uniform price spot market. *IEEE Trans. Power Syst.*, **2007**, *22*: 2152−2160.
154. Ciuprina, G.; Ioan, D.; Munteanu, I. Use of intelligent-particle swarm optimization in electromagnetics. *IEEE Trans. Magn.*, **2002**, *38*: 1037−1040.
155. Pontani, M.; Conway, B.A. Particle swarm optimization applied to space trajectories. *J. Guid., Control, Dyn.*, **2010**, *33*: 1429−1441.
156. Zhao, J.; Zhou, R. Particle swarm optimization applied to hypersonic reentry trajectories. *Chin. J. Aeronaut.*, **2015**, *28*: 822−831.
157. Cheng, G.H.; Jing, W.X.; Gao, C.S. Recovery trajectory planning for the reusable launch vehicle. *Aerosp. Sci. Technol.*, **2021**, *117*: 106965.

Citation: Fang, J.; Liu, W.; Chen, L.; *et al*. A survey of algorithms, applications and trends for particle swarm optimization. *International Journal of Network Dynamics and Intelligence*. 2023, 2(1): 24−50. doi: 10.53941/ijndi0201002

Publisher's Note: Scilight stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2023 by the authors. This is an open access article under the terms and conditions of the Creative Commons Attribution (CC BY) license https://creativecommons.org/licenses/by/4.0/.