A Convergence and Diversity Guided Leader Selection Strategy for Many-objective Particle Swarm Optimization

Lingjie Li[1], Yongfeng Li[1], Qiuzhen Lin[1]*, Zhong Ming[1]*, and Carlos A. Coello Coello[2,3]

[1]*College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, PR. China*

[2]*Computer Science Department, CINVESTAV-IPN, Mexico City, Mexico*

[3]*Basque Center for Applied Mathematics (BCAM) & Ikerbasque, Spain.*

Abstract:

Recently, particle swarm optimizer (PSO) is extended to solve many-objective optimization problems (MaOPs) and becomes a hot research topic in the field of evolutionary computation. Particularly, the leader particle selection (LPS) and the search direction used in a velocity update strategy are two crucial factors in PSOs. However, the LPS strategies for most existing PSOs are not so efficient in high-dimensional objective space, mainly due to the lack of convergence pressure or loss of diversity. In order to address these two issues and improve the performance of PSO in high-dimensional objective space, this paper proposes a convergence and diversity guided leader selection strategy for PSO, denoted as CDLS, in which different leader particles are adaptively selected for each particle based on its corresponding situation of convergence and diversity. In this way, a good tradeoff between the convergence and diversity can be achieved by CDLS. To verify the effectiveness of CDLS, it is embedded into the PSO search process of three well-known PSOs. Furthermore, a new variant of PSO combining with the CDLS strategy, namely PSO/CDLS, is also presented. The experimental results validate the superiority of our proposed CDLS strategy and the effectiveness of PSO/CDLS, when solving numerous MaOPs with regular and irregular Pareto fronts (***PF***s).

Keywords: Particle swarm optimization, Leader selection strategy, Many-objective optimization.

1. Introduction

Many-objective optimization problems (MaOPs) usually contain more than three objectives to be optimized simultaneously, which are extended from multi-objective optimization problems (MOPs) [1], as defined by

$$\begin{array}{ll} \text{minimize} & F(x) = (f_1(x), ..., f_m(x)), \\ \text{subject to} & x \in \Omega, \end{array} \tag{1}$$

where $x = (x_1, \ldots, x_n)$ is a decision vector from the search space Ω (n is the number of decision variables), and $F(x)$ defines m objective functions. Due to the conflicts often arising in different objectives, there is not a single optimal solution, but a set of trade-off solutions termed Pareto-optimal set (PS) for solving MOPs. The mapping of PS onto objective space is termed Pareto-optimal front (***PF***) [2]. During the last decades, many evolutionary algorithms (EAs) have become a popular and an effective heuristic approach to tackle MOPs [3]-[8] and real-world engineering optimization problems, such as the job shop-scheduling problem [9].

As another branch of heuristic, particle swarm optimization (PSO) algorithms have become a hot research area and owned excellent competitiveness for solving MOPs and MaOPs during the recent decades, due to their fast convergence and high computational efficiency. In [10], an adaptive local search strategy using a quasi-entropy index is embedded into the comprehensive learning PSO, which can intelligently determine the appropriate time to start the local search. In general, the position and velocity update for each particle are particularly crucial for the performance of PSOs. According to the principle of selecting the leader particles, most of existing PSOs can be roughly grouped into two main categories, i.e., the Pareto-based PSOs [11]-[21] and the decomposition-based PSOs [22]-[27].

In terms of the Pareto-based PSOs, the Pareto dominant relationship is utilized to select the leader particles. SMPSO [11], CMPSO [12] and OMOPSO [13] are three representative Pareto-based PSOs. After that, several new variants were presented to improve the performance of PSOs. For example, NMPSO [14] proposed a balanceable fitness estimation (BEF) method to provide a strong convergence pressure towards the true ***PF***, which takes both the convergence distance and diversity distance into consideration. In MaPSO [15], the leader particles were selected from a certain number of historical solutions that are obtained by using a scalar projection-based quality estimator, which shows a low computational complexity. AGMOPSO [16] proposed an adaptive gradient method and a self-adaptive flight parameters mechanism, which are expected to accelerate the convergence speed and maintain a good diversity, respectively. pccsAMOPSO [17] used a parallel cell coordinate system to detect the evolutionary environments during the PSO search process, which mainly includes the density, rank and diversity indicators as measured by using parallel cell distance, potential and distribution entropy, respectively. MOPSO/ESE [18] divided the evolutionary environment into the exploitation and exploration status, in which different leader particles are selected for different evolutionary environments. AMOPSO [19] presented a hybrid framework that combines the distribution entropy and population spacing information. Particularly, the distribution entropy is used to adaptively select the global leader particle, and the population spacing information is applied to dynamically adjust the flight parameters. MaOPSO/2s-pcss [20] proposed a two-stage strategy, where the convergence and diversity are separately emphasized at different stages by using a single-objective optimizer and a many-objective optimizer, respectively. Besides that, the parallel cell coordinate system in MaOPSO/2s-pcss is exploited to manage the diversity as well. MOPSO/DD [21] proposed a new scheme to rank the solutions in objective space, namely the dominant difference (DD) of a solution, which can demonstrate the dominant relationship in every dimension.

Regarding the decomposition-based PSOs, the decomposition approaches are embedded into standard PSOs and then the leader particles are selected according to the aggregation function values. MOPSO/D [22], dMOPSO [23] and D2MOPSO [24] are three well-known decomposition-based PSOs. After that, many PSOs were proposed to improve the performance of decomposition-based PSOs. For example, AWPSO [25] proposed a sigmoid-function-based weighting strategy to adaptively adjust the acceleration coefficients. MPSO/D [26] applied the decomposition method to ensure the diversity and adopted the crowding distance as the fitness value for operator selection. MS-PSO/D [27] utilized an element-based representation method and a constructive approach to coordinate with the property of problems.

Up to now, most of existing PSOs have been presented for solving MOPs in a low-dimensional objective space and shown very promising performance. However, they may encounter some difficulties when solving MaOPs and the studies of PSOs in a high-dimensional objective space are far away from enough. Regarding the performance on convergence, the PSO search ability is inefficient in most PSOs because they are difficult to approach the true ***PF*** in a high-dimensional objective space, due to the increase of dimensionality [21]. On the other hand, PSOs may also incur a rapid loss of diversity in a high-dimensional objective space [17]. To alleviate these issues and improve the performance of PSO in solving MaOPs, a convergence and diversity guided leader selection strategy, denoted as CDLS, is proposed in this article, in which the leaders of each particle are selected based on its corresponding condition of convergence and diversity. To summarize, the main contributions of this article are clarified as follows:

1) A convergence and diversity estimation strategy (CDES) is designed, which aims to detect the status of convergence and diversity for each particle at every generation. In this way, there are four different cases classified by using the proposed CDES, including ***Case* 1** (the particle shows a good convergence performance and is located in a crowded area), ***Case* 2** (the particle shows a good convergence performance and is located in a sparse area), ***Case* 3** (the particle shows a bad convergence performance and is located in a crowded area), and ***Case* 4** (the particle shows a bad convergence performance and is located in a sparse area).
2) In terms of selecting leader particles, a convergence and diversity guided leader selection (CDLS) strategy is proposed in this article for each particle, in which the global best particle and the local best particle are adaptively selected for each particle based on the status of its convergence and diversity. In other words, different leader particles are assigned for different scenarios intelligently, aiming to improve the robustness of PSO in a high dimensional objective space.
3) An efficient velocity update strategy using different leader particles is designed for different scenarios during the PSO search process, which is expected to improve the exploration ability of PSO on solving various kinds of MaOPs in a high-dimensional objective space.
4) The effectiveness of CDLS is validated by testing the performance of embedding our proposed strategy into three well-known PSOs (i.e., SMPSO [11], OMOPSO [13] and NMPSO [14]). In addition, a new PSO variant (called CDLS/PSO) is further presented for solving the WFG1-WFG9 test problems [28], the WFG41-WFG48 test problems [29] and the DTLZ1-DTLZ7 test problems [30] with 5 to 15 dimensions, in which a novel efficient space-division-based archive update strategy is also implemented to coordinate with the proposed CDLS strategy. The experimental results validate the superior performance of PSO/CDLS over several state-of-the-art competitors, including three competitive PSOs (i.e., MPSO/D [26], AGPSO [31] and MaPSO [15]) and three promising EAs for solving MaOPs (i.e., MaOEA/C [32], MaOEA/IT [33] and PAEA [34]).

The rest of this paper is organized as follows. Section 2 introduces a short review of some existing particle update strategies in PSOs and demonstrates our motivations. The details of the proposed CDLS strategy are explained in Section 3, while the experimental results and some relative discussions are presented in Section 4. Finally, our conclusions and future work are given in Section 5.

2. Related background and motivation

2.1 Particle swarm optimization

In 1995, the original PSO algorithm was first proposed by *Kennedy* and *Eberhart* as a global optimization method [35]. After that, there are many PSOs presented for solving continuous and unconstrained optimization problems. In general, PSO is an effective population-based optimization strategy, which simulates the movements of a flock of birds to find food.

In PSO, each particle *i* is associated with a velocity vector $v_i=[v_i^1,v_i^2,...,v_i^M]$ and a position vector $x_i=[x_i^1,x_i^2,...,x_i^M]$ to represent the current state, where *M* means the dimensionality of objective space. Then, each particle *i* will update both its velocity $v_i(t+1)$ and its position $x_i(t+1)$ to find the optimal solutions, according to a velocity update strategy, as follows:

$$v_i\left(t+1\right)=wv_i\left(t\right)+c_1r_1\left(pbest_i-x_i\left(t\right)\right)+c_2r_2\left(gbest_i-x_i\left(t\right)\right), \tag{2}$$

and the new position $x_i(t+1)$ for each particle *i* is updated, as follows:

$$x_i(t+1)=x_i(t)+v_i(t+1), \tag{3}$$

where *t* and *w* respectively represent the current generation and the inertia factor, c_1 and c_2 are two learning factors, r_1 and r_2 are two random numbers in the range (0, 1), $pbest_i$ and $gbest_i$ represent the personal best and the global best positions at current generation for the particle *i*, respectively.

2.2 Review of some existing particle swarm update strategies in PSO

In this section, some existing particle update strategies proposed in some popular PSOs [12], [15], [24]-[25] and [36], which adopt different leader selection strategies as well as different velocity update functions, are introduced in details.

1) *Particle Update Strategy in CMPSO* [12]:

CMPSO proposes a co-evolutionary technique in PSO search process, which applies a shared external archive to implement the information-sharing strategy. According to the information obtained from this shared external archive, a novel modified particle velocity update strategy is designed in CMPSO, as follows:

$$v_i\left(t+1\right)=wv_i\left(t\right)+c_1r_1\left(pbest_i-x_i\left(t\right)\right)+c_2r_2\left(gbest_i-x_i\left(t\right)\right)+c_3r_3\left(Abest_i-x_i\left(t\right)\right), \tag{4}$$

where $Abest_i$ is a non-dominated solution randomly selected from the shared external archive. In this way, CMPSO not only considers its corresponding personal experience and its swarm experience, but also takes the experience fetched from the shared external archive into consideration. Hence, the swarms can share their search information thoroughly via the archive.

2) *Particle Update Strategy in D^{2}MOPSO* [24]:

D^{2}MOPSO uses an external archive to save all the non-dominated particles, which are truncated based on the crowding distances in both objective and decision spaces. Then, the leader particle is selected from the archive to update the velocity, as follows:

$$v_i\left(t+1\right)=wv_i\left(t\right)+c_1r_1\left(pbest_i-x_i\left(t\right)\right)+c_2r_2\left(Lbest_i-x_i\left(t\right)\right), \tag{5}$$

where $Lbest_i$ indicates the leader particle selected from the archive that provides the best aggregation value constructed by using the decomposition approach [37] for the corresponding particle. Thus, D^{2}MOPSO incorporates the dominance relationship with the decomposition approach in PSO.

3) *Particle Update Strategy in AgMOPSO* [36]:

Inspired from the differential evolution (DE) operator [38] that has strong exploration ability, a novel velocity update method is designed in AgMOPSO, as follows:

$$v_i(t+1) = wv_i(t) + F_1 \cdot (pbest_i - x_i(t)) + F_2 \cdot (lbest_i - gbest_i), \tag{6}$$

where F_1 and F_2 respectively indicate the distance value d_1 and d_2 in penalty-boundary-intersection method [32]. That is to say, F_1 is set to a high value if the particle is far away from the current sub-problem and F_2 is set to a high value if the particle is far away from the ideal point. Moreover, for the third part in (6), the differential vector is similar to the 'DE/rand/1' operator [38], which aims to improve the disturbance ability. Here $lbest_i$ means the local best particle that is selected from the corresponding neighborhood. In this way, AgMOPSO not only has the search pattern of PSO, but also inherits the effective search ability of DE.

4) *Particle Update Strategy in NMPSO* [14]:

In order to provide another search direction and make more disturbances during the PSO search process, a novel search direction from the personal best particle to the global best one is inserted in NMPSO. Thus, the velocity update function in NMPSO is formulated as follows:

$$v_i(t+1) = wv_i(t) + c_1r_1(pbest_i - x_i(t)) + c_2r_2(gbest_i - x_i(t)) + c_3r_3(gbest_i - pbest_i), \tag{7}$$

where $gbest_i$ is randomly selected from the top 10% individuals in the external archive with the better BFE (balanceable fitness estimation) value. In this way, the embedded search direction helps to guide the swarm to search toward the global best particle, which is expected to provide a strong convergence pressure during the PSO search process.

5) *Particle Update Strategy in MaPSO* [15]:

In MaPSO, the leader particle is selected from a number of historical solutions by using the scalar projections, in which the historical solutions are used to record the potential search direction. Moreover, to avoid negative guidance caused by the leader particle, a sign function, denoted as $\mathrm{sgn}(bsp_i)$, is also used to adaptively adjust the search direction for each parctile. The particle update function in MaPSO is constructed as follows:

$$v_i(t+1) = wv_i(t) + c_1r_1 \cdot \mathrm{sgn}(bsp_i) \cdot (Lbest_i - x_i(t)), \tag{8}$$

$$\text{with } \mathrm{sgn}(bsp_i) = \begin{cases} 1 & if\ bsp_i \geq 0 \\ -1 & if\ bsp_i < 0 \end{cases}, \tag{9}$$

where $Lbest_i$ is selected from the historical solutions that are closest to the ***PF*** along the corresponding reference vector and $\mathrm{sgn}(bsp_i)$ is applied to determine the search direction for each particle. Specifically, the leader is closer to the ***PF*** than the current particle when $bsp_i > 0$. In this situation, the particle will search in the same direction. Otherwise, the particle will search in the negative direction. In this way, this method helps to pull the whole swarm toward the true ***PF*** quickly.

6) *Particle Update Strategy in AWPSO* [25]:

In AWPSO, the acceleration coefficients in velocity update function are adaptively adjusted by using the designed sigmoid-function-based weighting strategy, which considers both the distances from the particle to its personal best particle and from the particle to the global best particle, as follows:

$$v_i(t+1) = wv_i(t) + c_{d_{pi}}(t)\cdot r_1 \cdot d_{pi}(t) + c_{d_{gi}}(t)\cdot r_2 \cdot d_{gi}(t), \tag{10}$$

where $d_{pi}(t)$ and $d_{gi}(t)$ represent the distance from the particle i to its *pbest* and *gbest* at t-th generation, respectively. $c_{d_{pi}}(t)$ and $c_{d_{gi}}(t)$ mean the acceleration constants that are respectively determined by $d_{pi}(t)$ and $d_{gi}(t)$ with the proposed sigmoid function. More details about the sigmoid function can be found in [25]. In this way, this method can facilitate relatively fast exploitation in the search space.

2.3 Our motivations

In this section, the performance of these traditional PSOs for solving MaOPs in a high-dimensional objective space is investigated. Some experiments were performed on WFG43 [33], WFG9 [34] and DTLZ4 [35] test problems with 3 to 15 objectives. Particularly, seven PSOs, including CMOPSO [39], dMOPSO [23], MMOPSO [40], MOPSOCD [41], SMPSO [11], MaPSO [15] and NMPSO [14], and two competitive many-objective EAs (MaOEAs) (NSGA-III [42] and VaEA [43]) were included for experimental comparisons. To have a visual observation on the performance of each algorithm, the average HV [44] values that are obtained by each compared algorithm after 30 independent runs were plotted in Fig. 1. The parameters in these compared algorithms were set as suggested in the corresponding references [14]-[15], [23], [39]-[43], which can also be found in Section 4.

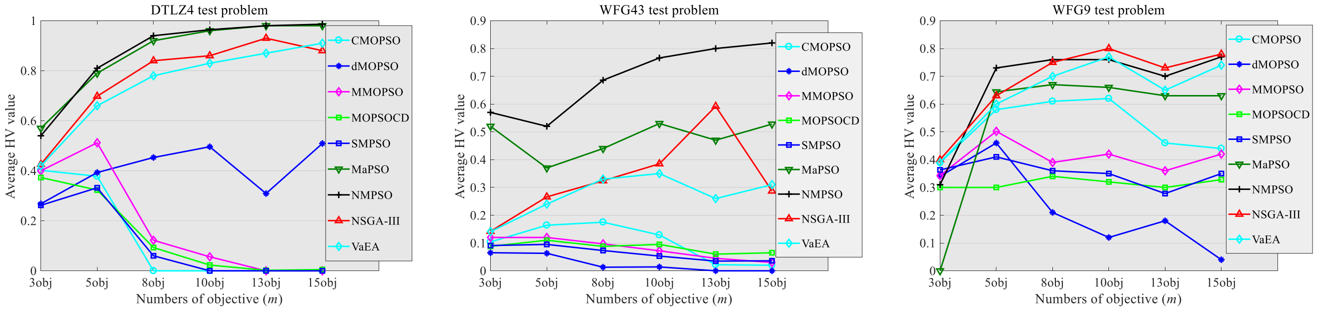

Fig.1 The HV values of seven PSOs and two EAs for solving DTLZ4, WFG43 and WFG9 with 3 to 15 objectives.

As we can learn from Fig. 1, the performance of all the adopted PSOs is obviously deteriorated with the increased numbers of objectives, except for NMPSO and MaPSO. These empirical results provided in Fig. 1 demonstrate the inefficiency of most traditional PSOs when solving MaOPs in a high-dimensional space.

Here, we further analyze the main reasons behind this theoretically. As reviewed in Section 2.2, the main difference between these compared PSOs is their corresponding adopted leader particle selection (LPS) strategy during the PSO search process. For most traditional PSOs, their LPS strategies only use some simple selection criteria. Specifically, SMPSO, CMOPSO and OMOPSO apply the dominant relationship to select the leader particles [11]-[13]. However, the dominant relationship become invalid with the increase of objective numbers, as most solutions in population may be mutually non-dominated in a high-dimensional objective space. Therefore, these three dominance-based PSOs show inefficiency for solving MaOPs. Second, for decomposition based PSOs (i.e., dMOPSO, MOPSOCD, MMOPSO), they adopt the aggregated function values constructed by decomposition approaches [22]-[24] to select the leader particles.

However, it is difficult to specify a uniform set of weight vectors in a high-dimensional objective space, thereby may lose the diversity when solving MaOPs. In order to improve the performance of PSO in solving MaOPs, two promising PSO variants (i.e., NMPSO and MaPSO) are presented recently, which adopt more effective LPS strategies to enhance the search capabilities of PSOs in a high-dimensional objective space. Specifically, NMPSO [14] uses a BFE metric to guide the selection of leader particles, which provides a strong convergence pressure, while MaPSO [15] adopts a scalar projection-based quality estimator to select the leader particles, which can speed up the convergence. However, the BFE metric in NMPSO needs a high computational complexity, while the scalar projection-based quality estimator in MaPSO is ineffective for solving some MaOPs with irregular ***PF***s, like WFG43 with a concave ***PF*** as shown in Fig. 1. Hence, the search capabilities of PSOs in a high-dimensional objective space are still far from satisfactory. The design of an effective LPS strategy to solve MaOPs deserves further study, as an appropriate leader particle selected by the LPS strategy can provide a promising search direction that guide the population to approximate to the true ***PF***.

To summarize, there are two main challenges for most traditional PSOs to optimize MaOPs in a high-dimensional objective space. One of the challenges is the loss of sufficient selection pressure to approximate the true ***PF***s [14]. The other one is the premature convergence occurred in most traditional PSOs [45]. To address these issues and improve the robustness of PSO in solving various MaOPs, a convergence and diversity guided leader particle selection (CDLS) strategy is proposed in this paper. The main idea behind CDLS is to adaptively select the most appropriate leader particles for each particle based on its corresponding condition, which takes the convergence status and diversity information into consideration. More details of our proposed CDLS are introduced in the next section.

3. The proposed CDLS strategy

In this section, the details of the proposed CDLS strategy are introduced. At first, a general framework based on the proposed CDLS strategy is described in Section 3.1. Then, the proposed CDLS strategy is introduced in Section 3.2, which mainly includes 1) the proposed convergence and diversity estimation strategy (CDES), 2) the process of leader particle selection, and 3) the used particle update strategy.

3.1 The complete framework of the CDLS strategy

The pseudo-code of the complete framework of the CDLS strategy for PSOs is given in **Algorithm 1**, where *fes* and *MaxFes* indicate the current evaluation numbers and the maximum evaluation numbers, respectively. *N* is the population size. Lines 1 to 6 belong to the initialization process. *A* is initialized as an empty archive firstly. Then, the particle swarm *P* is initialized in lines 2 to 6. Specifically, the position information x_i and the velocity v_i for each particle *i* are initialized in line 3 and the objective values of each particle are evaluated in line 4. After that, the particle swarm *P* is used to run the main loop, which mainly includes two parts. The first part belongs to the proposed CDLS strategy as shown in line 9, which is introduced in Section 3.2. After that, the second part belonging to the general part of PSO, which mainly includes the population update process and evolutionary search process, is performed on particle swarm successively in lines 11-13. The main loop is terminated when *fes* reaches the predefined *MaxFes*. Finally, all the particles in *P* are outputted as the optimal solution set.

Algorithm 1 The complete framework of CDLS

Input: *MaxFes*, *N*

Output: the final *P*

```
initialize $A=\varnothing$, $P=\{p_1, p_2,..., p_N\}$;
for each $p_i$ in particle $P$
    randomly initialize the position $x_i$ and set $v_i$ =0 for $p_i$;
    evaluate the objective value of $p_i$;
    set $pbest_i$ =$p_i$;
end for
while $fes \le$ $MaxFes$
    //The part of CDLS strategy:
    $P'$=CDLS ($P$); // Algorithm 2
    //The general part of PSO:
    $A$=Population update ($P$, $P'$);
    $A'$=Evolutionary search ($A$);
    $P$=Population update ($A$, $A'$);
    $fes$=$fes$+$N$;
end while
```

3.2 The proposed CDLS strategy

Algorithm 2 The proposed CDLS strategy

Input: the current particles *P*

Output: the updated particles *P'*

```
1   CDES ($P$); //Algorithm 3
2   for each particle $p_i$ in $P$
3       select $gbest_i$ , $lbest_i$ for $p_i$ based on its corresponding Case;
4       update the velocity $v_i$ of $p_i$ by using (18);
5       update the position $x_i$ of $p_i$ by using (3);
6       evaluate the objective value of $p_i$;
7       if $pbest_i$ is dominated by $p_i$
8           set $pbest_i$ = $p_i$;
        else if $pbest_i$ is mutally mutually non-dominated with $p_i$
            $pbest_i$ is replaced by $p_i$ with a probability of 0.5;
9       end if
10  end for
```

The pseudo-code of the proposed CDLS strategy is given in **Algorithm 2**. At the beginning of CDLS, CDES is performed on the particle swarm *P* in line 1, which divides all the particles in *P* into four different categories. After that, as shown in line 3, different leader particles are determined and then assigned to guide different particles based on their corresponding statuses. Then, the PSO search process is performed on each particle p_i in lines 4-9, which updates its velocity and position information, respectively. The proposed CDES, the proposed CDLS strategy, and the used particle update strategy will be introduced in details successively, as follows.

3.2.1 The convergence and diversity estimation strategy

Algorithm 3 CDES procedure

```
1   for each particle $p_i$ in $P$
2       $Ed_i \Leftarrow$ using (11);
3       $Ld_i \Leftarrow$ using (15);
4   end for
5   $AvgEd_t$, $AvgLd_t \Leftarrow$ using (16) and (17);
6   //CDDM:
7   for each particle $p_i$ in $P$
8       Case 1 $\Leftarrow Ed_i < AvgEd_t \wedge Ld_i < AvgLd_t$;
9       Case 2 $\Leftarrow Ed_i < AvgEd_t \wedge Ld_i > AvgLd_t$;
10      Case 3 $\Leftarrow Ed_i > AvgEd_t \wedge Ld_i < AvgLd_t$;
11      Case 4 $\Leftarrow Ed_i > AvgEd_t \wedge Ld_i > AvgLd_t$;
12  end for
```

The pseudo-code of the proposed CDES is given in **Algorithm 3**. First, the status of the convergence and diversity for each particle is estimated in lines 1-4. Specifically, the status of the convergence for each particle is detected, which is reflected by using the Euclidean distance Ed_i from the particle p_i to the ideal point Z^*, calculated by:

$$Ed_i = \sqrt{\sum_{k=1}^{M}\left(f_k^{'}\left(p_i\right) \text{-} Z_k^*\right)^2}, i \in [1, N], \tag{11}$$

where N is the size of particles, $Z^* = \left\{Z_1^*, ..., Z_M^*\right\}$ indicates the ideal point ($Z_k^* = \min\left\{f_k(x)\right\}, k \in M$, and M is the objective number). Each objective of particle is normalized by using the ideal point Z^* and the nadir point $Z^{nadir} = \left\{Z_1^{nadir}, ..., Z_M^{nadir}\right\}$ with $Z_k^{nadir} = \max\left\{f_k(x)\right\}, k \in M$. Here, the normalized objective is denoted as $f_k^{'}\left(p_i\right)$ in (11), obtained as follows:

$$f_k^{'}\left(p_i\right) = \frac{f_k\left(p_i\right) - Z_k^*}{Z_k^{nadir} - Z_k^*}, k \in [1, M], \tag{12}$$

The normalized process can eliminate the effect of different amplitudes on multiple objectives and does not need any information from the true ***PF***s of problem [46]. Please note that the value of each $f_k^{'}\left(p_i\right)$ is ranging from 0 to 1 after normalization.

Second, the status of local density for each particle is estimated by using the diversity distance measured by the Euclidean distances between their projected points on the unite hyperplane, which has been widely used in MaOEAs [47]-[48] to measure the density of regions, defined by:

$$Dd\left\langle p_i, p_j\right\rangle = \left\|\lambda\left(p_i\right) - \lambda\left(p_j\right)\right\|_2, \tag{13}$$

$$\text{with } \lambda\left(p_i\right) = \frac{1}{\sum_{k=1}^{M} f_k^{'}\left(p_i\right)} \times F^{'}\left(p_i\right), \tag{14}$$

where $i \in [1, N], j \in [1, T] \wedge i \neq j$, $F^{'}(p_i) = (f_1^{'}(p_i), f_2^{'}(p_i), ..., f_M^{'}(p_i))^T$ means the normalized objective vector using (12), $\|\cdot\|_2$ indicates the 2-norm of a vector. T is the size of neighbor set for the particle p_i, where the neighbor set of p_i includes T solutions with the closest diversity distance in (13) to itself. Note that T is set as $0.1 \times N$ in this paper as suggested in [49]-[50]. In addition, due to page limitations, a parameter sensitivity analysis of T is given in Table S1 of supplementary file. In this way, the local density value for each particle, denoted as Ld_i, is calculated to reflect its corresponding local density status, defined below:

$$Ld_i\big|_{i=1}^{N} = \min\left\{Dd\left\langle p_i, p_1\right\rangle, Dd\left\langle p_i, p_2\right\rangle, \ldots, Dd\left\langle p_i, p_T\right\rangle\right\}, \tag{15}$$

After that, the average Euclidean distance and the average density value of particle swarm at the current t generation are calculated in line 5, denoted as $AvgEd_t$ and $AvgLd_t$, respectively, which are formulated as follows:

$$AvgEd_t = \frac{\sum_{i=1}^{N} Ed_i}{N}, \tag{16}$$

$$AvgLd_t = \frac{\sum_{i=1}^{N} Ld_i}{N}, \tag{17}$$

where t and N respectively indicate the current generation and the population size. Then, the Euclidean distance Ed_i and the local density value Ld_i for each particle are used to compare with $AvgEd_t$ and $AvgLd_t$, respectively.

By this way, as shown in lines 7-12 of **Algorithm 3**, four different statuses are classified by using the proposed CDES. They are respectively ***Case* 1**: the particle p_i has a good performance on convergence and is located in a crowded area (i.e., $Ed_i < AvgEd_t \wedge Ld_i < AvgLd_t$); ***Case* 2**: the particle p_i shows a good performance on convergence and is located in a sparse area (i.e., $Ed_i < AvgEd_t \wedge Ld_i > AvgLd_t$); ***Case* 3**: the particle p_i has a poor performance on convergence and is located in a crowded area (i.e., $Ed_i > AvgEd_t \wedge Ld_i < AvgLd_t$); ***Case* 4**: the particle p_i shows a poor performance on convergence and is located in a sparse area (i.e., $Ed_i > AvgEd_t \wedge Ld_i > AvgLd_t$). In order to have a more intuitive observation and easy understanding of these four different statuses that are classified by using our proposed CDES, Fig. 2 gives a simple example to display each status in a bi-objective space.

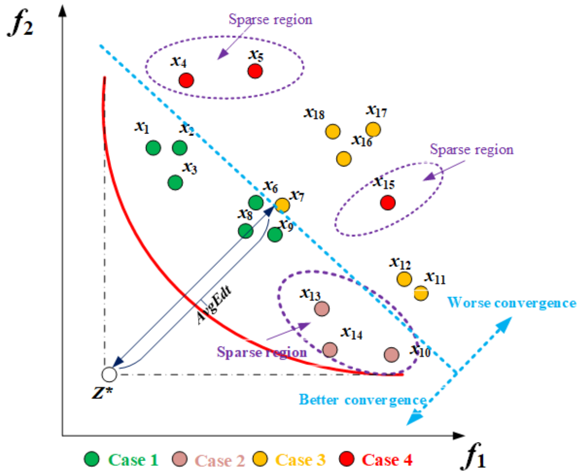

Fig. 2 Example with four statuses for each particle in a bi-objective space.

3.2.2 The proposed leader selection process

After the CDES process is completed, as shown in line 3 of **Algorithm 2**, the leader selection process is performed, which assigns different leader particles (i.e., the global best particle $gbest_i$ and the local best particle $lbest_i$) for each particle based on its corresponding status. Summarily, the leader particles are grouped into two categories, including the convergence-best leaders (i.e., $gbest_i^C$ and $lbest_i^C$) and the diversity-best leaders (i.e., $gbest_i^D$ and $lbest_i^D$). Particularly, regarding the global leader selection, the particle with the smallest Ed_i value using (11) is selected as $gbest_i^C$, while the particle with a maximum local density value Ld_i using (13)-(15) is selected as $gbest_i^D$. For local best leader selection, the particle with the smallest sum value of objectives among the *T* neighbors for particle p_i is selected as $lbest_i^C$, while the particle with the largest local density value Ld_i using (13)-(15) among the neighbor set of p_i is selected as $lbest_i^D$. The details of the leader selection process for four different statuses are introduced as follows:

- ***Case* 1**: Regarding the selection of $gbest_i$, the diversity should be more considered when the particle p_i shows a good performance on convergence. Hence, $gbest_i^D$ is selected as the global particle for such a particle, which is used to enhance the performance on diversity. On the other hand, regarding the selection of $lbest_i$, it is selected from the *T*-closest neighbors of p_i that is formulated according to the diversity distance in (13). For the particle p_i that is located in a crowded area, it is reasonable to pay more attention on the performance on its convergence. Thus, $lbest_i^C$ is selected as the local best particle to guide the search direction.
- ***Case* 2**: Similar to ***Case* 1**, $gbest_i^D$ is selected as the global best particle to improve the performance on diversity. Regarding the selection of $lbest_i$, more attention should be paid on maintaining diversity when the particle is located in a sparse region. Hence, $lbest_i^D$ is selected as $lbest_i$ for the particle p_i that is located at a sparse region, which aims to improve its diversity.
- ***Case* 3**: Regarding the selection of $gbest_i$, more attention should be paid on the convergence when the particle p_i shows a poor performance on convergence. As a result, $gbest_i^C$ is selected as $gbest_i$ to accelerate the convergence speed of p_i. Regarding the selection of $lbest_i$, $lbest_i^C$ is selected from the neighbor set as $lbest_i$ for the particle that is located in a crowded region, which is expected to speed up its convergence speed.
- ***Case* 4**: Similar to ***Case* 3**, $gbest_i^C$ is selected as $gbest_i$ to accelerate the convergence speed for the particle that shows a poor performance on convergence. Regarding the selection of $lbest_i$, $lbest_i^D$ is selected for the particle located in a sparse region, which aims to improve its diversity.

To conclude, different leader particles (i.e., $gbest_i^D$ or $gbest_i^C$, $lbest_i^D$ or $lbest_i^C$) are adaptively assigned for each particle according to its corresponding status on both convergence and diversity. In order to have an intuitive observation and comprehensive understanding of leader selection process, the selection logical structure is given in Fig. 3.

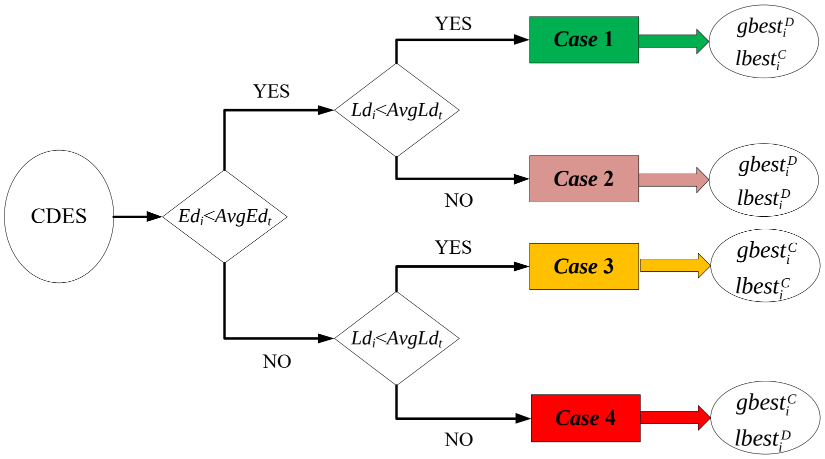

Fig. 3 The process of leader particle assignment for each status.

3.2.3 The used particle update strategy

After the leader selection process is completed, the selected leader particles are used to perform the PSO search in lines 4-9 of **Algorithm 2**. First, similar to some existing PSOs [14], [36], the velocity of the particle p_i in this paper is updated by using a classic velocity update function, which applies the personal-best, global-best and personal-best particles to guide the search direction of particle p_i simultaneously, defined as follows:

$$v_i\left(t+1\right)=wv_i\left(t\right)+c_1r_1\left(pbest_i-x_i\left(t\right)\right)+c_2r_2\left(gbest_i-x_i\left(t\right)\right)+c_3r_3\left(lbest_i-x_i\left(t\right)\right), \tag{18}$$

where t and w respectively indicate the generation number and the inertial weight for particle p_i; c_1, c_2 and c_3 are three learning factors; r_1, r_2 and r_3 are three random numbers ranging from 0 to 1. The selection of $gbest_i$ and $lbest_i$ has been introduced above, which is determined by the status of convergence and diversity of each particle. Regarding the selection of personal best particle, the particle that can provide the best result for i-th particle is selected as $pbest_i$. Then, the position of particle p_i is updated by using Eq. (3) and the objective value of p_i is also reevaluated in line 6. In addition, $pbest_i$ is updated in lines 7-9. Normally, if $pbest_i$ is dominated by the new solution generated by particle p_i, $pbest_i$ will be replaced by this new solution. If $pbest_i$ is mutually non-dominated with this new solution, $pbest_i$ will be kept the same with a probability of 0.5 and be replaced by the new solution with the same probability.

4. Experimental results

4.1 Benchmark problems

In our experimental studies, twenty-five benchmark problems, including the WFG1-WFG9 test problems [28], the WFG41-WFG48 test problems [29] and the DTLZ1-DTLZ7 test problems [30], were used to evaluate the performance of the proposed algorithm.

Particularly, these used test problems mentioned above have different ***PF***s. For the WFG test problems, WFG1 and WFG 2 have convex ***PF***s. In particular, WFG1 is designed with a mixed ***PF*** while WFG2 has a disconnected ***PF***. WFG3 has a linear and degenerate ***PF***. WFG4-WFG9 all have concave ***PF***s. Moreover, the multi-modality of WFG4 has larger hill sizes than that of the non-separable reduction of WFG9. WFG1 and WFG7 are both separable and uni-modal; For the DTLZ test problems, DTLZ1 has a linear and multi-modal ***PF***, DTLZ2-DTLZ6 have concave ***PF***s, and DTLZ7 has a disconnected ***PF***. Particularly, DTLZ3 is a multi-modal problem, DTLZ4 is a biased problem, and DTLZ5-DTLZ6 are two degenerated problems. For the WFG4X test problems, WFG41 and WFG43 have concave ***PF***s, while WFG42, WFG44 and WFG47 all have convex ***PF***s. Moreover, WFG44 has a strong convex ***PF*** and the ***PF*** of WFG47 is disconnected. WFG45 has a mixed ***PF*** and the ***PF*** of WFG46 is a hyper-plane. Hence, the experimental comparison on these adopted problems can effectively assess the performance of all the compared algorithms.

The numbers of objectives m for each problem were set as $\left\{5,8,10,13,15\right\}$. Regarding the number of decision variables, for the WFG41-WFG48 and WFG1-WFG9 test problems, the decision variables are composed by k position-related parameters and l distance-related parameters. As learned from [26]-[27], k and l were respectively set to $2\times\left(m-1\right)$ and 20. For the DTLZ1-DTLZ7 test problems, the decision variables were set as $n = m + k - 1$. As introduced in [28], the values of *k* were set to 5 for DTLZ1, 10 for DTLZ2-DTLZ6, and 20 for DTLZ7, respectively.

4.2 Performance metrics

In our experiment, we use two widely used performance metrics to evaluate the performance of each compared algorithm, including the hyper-volume (HV) metric [44] and the inverted generation distance (IGD) [51], which can reflect the quality of both convergence and diversity for the final solution set [52]. Specifically, HV indicates the hyper-volume of the space that is constructed by the non-dominated solutions and the reference point. A larger HV value indicates a better approximation to the true ***PF***. In our experimental studies, the reference point for calculating the HV value was set as suggested in [53], in which all the objective values of the final solution were normalized first and then the reference point was set as $(1.0,1.0,...,1.0)$. Particularly, in order to improve the computational efficiency, the Monte Carlo simulation method [54] with 10^7 sampling points was adopted to approximately calculate the HV values when solving the problems with more than 8 objectives (i.e., $m \in \{8,10,13,15\}$). IGD indicates the average Euclidean distance from the obtained final solutions to the points that are distributed along the true ***PF*** in the objective space. A lower IGD value means a better performance. Moreover, in order to ensure a statistically sound conclusion, the Wilcoxon rank sum test with a 0.05 significance level and the Wilcoxon signed-rank test using the platform ***KEEL*** [55] were also used in the experimental analysis.

4.3 Algorithms for comparison

In our experiments, the proposed CDLS was embedded into three well-known PSOs, including SMPSO [11], OMOPSO [13] and NMPSO [14], to validate the superiority and effectiveness of the proposed CDLS when compared with their original algorithms. In addition, three competitive PSOs (MPSO/D [26], AGPSO [31] and MaPSO [15]) and three promising MaOEAs (MaOEA/C [32], MaOEA/IT [33] and PAEA [34]) were also compared for performance verification. A brief introduction of each compared algorithm is given as follows.

1) SMPSO [11]: SMPSO presents a speed-constrained velocity update strategy, which aims to control the velocity for each particle.
2) OMOPSO [13]: OMOPSO is a Pareto-based PSO, which uses a crowding factor to filter out the list of available leaders.
3) NMPSO [14]: NMPSO proposes a balanceable fitness estimation method and a novel velocity update function, which takes the diversity and convergence into consideration more reasonably.
4) MPSO/D [26]: MPSO/D is a decomposition-based PSO, in which the diversity is guaranteed by using the decomposition method and the convergence pressure is strengthened by using the crowding distance-based fitness values that is used to determine the global best particle.
5) AGPSO [31]: AGPSO presents an angular-distance-based PSO, in which an effective velocity update strategy is designed to enhance the search intensity around the selected leader particles.
6) MaPSO [15]: MaPSO selects the leader particles from a certain number of historical solutions, which is based on an efficient scalar projection-based quality estimator.

7) MaOEA/C [32]: MaOEA/C is a clustering-based many-objective optimization algorithm, in which the individuals showing high similarities on the vector angles are gathered into the same cluster.
8) MaOEA/IT [33]: MaOEA/IT addresses the convergence and diversity into two independent and sequential stages, in which a novel non-dominated dynamic weight aggregation method is designed to find the optimal solutions.
9) PAEA [34]: PAEA is a decomposition-based many-objective evolutionary algorithm that uses adaptive search directions and two reference points simultaneously.

Please note that the proposed CDLS strategy is implemented under the jMetal framework [56]. Other compared algorithms were all implemented under the jMetal framework, except that MaOEA/IT [33] and PAEA [34] were implemented in the PlatEMO framework [57]. Moreover, all the compared algorithms mentioned above were run on a personal computer with an Intel ® Core ™ i7-6700 CPU, 3.40 GHZ (processor), and 20 GB (RAM).

4.4 Parameters settings for the compared algorithms

1) *Population size* (*N*): Regarding the population sizes *N* for different numbers of objectives, the two-layer generation approach with the simplex-lattice designed factor *H* designed in [58] was adopted in this paper. The numbers of weight vectors were respectively set to 210, 240, 275, 182 and 240 for 5-, 8-, 10-, 13- and 15-objective test problems. Please note that the population sizes *N* were set the same as the numbers of weight vectors.
2) *Number of function evaluations* (*MaxFes*): Regarding the termination condition, all the compared algorithms are terminated when the predefined maximum number of generations *MaxGen* is reached. The settings of *MaxGen* for solving MaOPs with different numbers of objectives were 600 (for 5 objectives), 800 (for 8 objectives), 1000 (for 10 objectives), 1200 (for 13 objectives) and 1500 (for 15 objectives), respectively. Thus, the maximum function evaluations (*MaxFes*) for different numbers of objectives were determined by $MaxFes = N \times MaxGen$.
3) *Crossover and Mutation*: Two widely used evolutionary operators, i.e., simulated binary crossover (SBX) [59] and polynomial mutation (PM) [60], were implemented in all the compared algorithms. Based on the suggestion in [32]-[34], the probabilities of SBX and PM were respectively set to 1.0 and 1/*n* (*n* is the number of decision variables). The distribution indexes of SBX and PM were all set to 20.

4.5 Results of embedding CDLS strategy into three PSOs

In this part, the CDLS strategy was embedded into three well-known PSOs (SMPSO [11], OMOPSO [13] and NMPSO [14]), forming three new variants, denoted as SMPSO/CDLS, OMOPSO/CDLS, NMPSO/CDLS, respectively. Here, three comparisons of SMPSO/CDLS versus SMPSO, OMOPSO/CDLS versus OMOPSO, and NMPSO/CDLS versus NMPSO were run on three widely used benchmark problems (i.e., DTLZ, WFG and WFG4X test problems) with 5 to 15 objectives.

4.5.1 Comparison results on the WFG41-WFG48 test problems

Table S2 of supplementary file provides the average HV values and the variance for the WFG41-WFG48 test problems with 5 to 15 objectives in 30 independent runs. Some conclusions can be drawn from the HV values listed in Table S2. As shown in the second-to-last row of the table, these three new variants embedded with the proposed CDLS strategy showed an obvious superiority over their original versions. Specifically, SMPSO/CDLS, OMOPSO/CDLS and NMPSO/CDLS respectively obtained the best results in 22, 38 and 31 out of 40 cases, while their corresponding original versions only showed the best performance in 18, 0 and 9 out of 40 cases. Moreover, from the one-by-one comparisons in the last row of Table 1, SMPSO/CDLS, OMOPSO/CDLS and NMPSO/CDLS also performed better than their competitors in 11, 37 and 13 out of 40 cases, while they only underperformed respectively in 5, 0, and 1 out of 40 cases. The main difference between the compared algorithms is the selection of leader particles. Hence, the experimental results listed in Table S2 validated the effectiveness of the proposed CDLS strategy, because it can improve the performance of the original PSOs in solving WFG4X test problems. The similar conclusion can be drawn from the IGD results summarized in Table S3 of supplementary file, as our proposed CDLS strategy can significantly improve the performance of the original PSO variants (i.e., SMPSO, OMOPSO and NMPSO) in most WFG4X test problems ranging from 5 to 15 objectives.

4.5.2 Comparison results on WFG1-WFG9 test problems

Table S4 of supplementary file summarizes the comparison results based on the HV values for WFG1-WFG9 test problems from 5 to 15 objectives. Some conclusions can also be learned from the HV values listed in Table S4. As listed in the second-to-last row of the table, these three new variants equipped with the proposed CDLS strategy showed an obvious superiority over its original versions without using the CDLS strategy. Specifically, SMPSO/CDLS, OMOPSO/CDLS and NMPSO/CDLS respectively obtained the best results in 29, 36 and 41 out of 45 cases, while their original versions only showed the best performance in 16, 9 and 4 out of 45 cases, respectively. Moreover, from the one-by-one comparisons in the last row of Table S4, SMPSO/CDLS, OMOPSO/CDLS and NMPSO/CDLS also performed better than their original versions in 17, 21 and 24 out of 45 cases, while they only performed worse than their original versions in 9, 3 and 0 out of 40 cases. The main difference between these competitors is the proposed CDLS implemented in the PSO search process. Hence, the HV results summarized in Table S4 have further verified the superiority of the proposed CDLS strategy on solving these WFG problems. In addition, due to page limitations, Table S5 of supplementary file provides the average IGD results obtained by each compared algorithm, which further demonstrate that our proposed CDLS strategy can promote the performance of conventional PSOs in solving WFG test problems in a high-dimensional objective space.

4.5.3 Comparison results on the DTLZ1-DTLZ7 test problems

Table S6 of supplementary file provides the experimental results according to the HV values obtained by each compared algorithm on tackling DTLZ1-DTLZ7 benchmark problems with 5 to 15 objectives. Some conclusions can be drawn from the results summarized in Table S6. As listed in the second-to-last row of this table, these three new variants equipped with the proposed CDLS strategy showed obvious superiority over their original versions. Specifically, SMPSO/CDLS, OMOPSO/CDLS and NMPSO/CDLS can respectively obtain the best results in 24, 23 and 19 out of 35 cases, while their corresponding original versions only showed the best performance in 10, 10 and 16 out of 35 cases. On the other hand, regarding the one-by-one comparisons in the last row of Table S6, SMPSO/CDLS, OMOPSO/CDLS and NMPSO/CDLS also respectively performed better than their original versions in 16, 19 and 11 out of 35 cases, while they only underperformed in 7, 6 and 5 out of 35 cases, respectively. Furthermore, the IGD results listed in Table S7 of supplementary file have also validated the superiority of three new PSO variants (i.e., SMPSO/CDLS, OMOPSO/CDLS and NMPSO/CDLS) embedded with our designed CDLS strategy over their corresponding original PSO variants (i.e., SMPSO, OMOPSO and NMPSO). In summary, the experimental results in Tables S6 –S7 of supplementary file have verified the effectiveness of the proposed CDLS strategy for solving DTLZ benchmark problems.

4.5.4 Further discussions and analyses on the overall performance of compared algorithms

Table 1

Summary of significance test between three PSOs and their enhanced version combined with CDLS based on the HV values

Comparisons based on		SMPSO/CDLS vs. SMPSO				OMOPSO/ CDLS vs. OMOPSO				NMPSO/ CDLS vs. NMPSO			
		–	~	+	*p*-value	–	~	+	*p*-value	–	~	+	*p*-value
Test Suites	DTLZ	**16**	**12**	**7**	**0.0432**	**19**	**10**	**6**	**0.0150**	**11**	**19**	**5**	**0.0193**
	WFG	**17**	**19**	**9**	**0.0570**	**21**	**21**	**3**	**0.0000**	**24**	**21**	**0**	**0.0000**
	WFG4X	**11**	**24**	**5**	1.0000	**37**	**3**	**0**	**0.0000**	**13**	**26**	**1**	**0.0000**
No. (m)	*m*=5	**19**	**5**	**0**	**0.0000**	**16**	**6**	**2**	**0.0007**	**8**	**16**	**0**	**0.0104**
	m=8	**8**	**12**	**4**	**0.9426**	**15**	**9**	**0**	**0.0006**	**7**	**17**	**0**	**0.0109**
	m=10	**8**	**11**	**5**	1.0000	**14**	**8**	**2**	**0.0005**	**17**	**5**	**2**	**0.0000**
	m=13	3	14	7	1.0000	**16**	**6**	**2**	**0.0008**	**6**	**15**	**3**	**0.0121**
	m=15	**6**	**13**	**5**	1.0000	**17**	**5**	**2**	**0.0004**	**8**	**16**	**0**	**0.0031**
	All	**44**	**55**	**21**	**0.0441**	77	**34**	**9**	**0.0000**	**48**	**66**	**6**	**0.0000**

Table 1 provides the significance test results based on the average HV values for each comparison. In order to ensure a statistically sound conclusion, the Wilcoxon rank-sum test with a 0.05 significance level and the Wilcoxon signed-rank test using the ***KEEL*** platform [55] were applied, which show statistically significant differences on the HV results. In Table 1, the labels "-", "~" and "+" respectively mean the comparison times that the results of these three new variants (SMPSO/CDLS, OMOPSO/CDLS and NMPSO/CDLS) are better than, similar to, worse than their corresponding original algorithms (SMPSO, OMOPSO, and NMPSO). Besides that, the column "*p*-value" summarizes the asymptotic *p*-value obtained by using the Wilcoxon signed-rank test on the ***KEEL*** platform. Please note that a *p*-value closer to zero indicates that there are more significant differences between the competitors. As learned from the HV results listed in Table 1, the superiority of our proposed CDLS strategy has been further validated, which can obviously improve the performance of the original PSOs.

Regarding the results summarized in Table 1 for all test problems, some conclusions can be easily drawn. When considering all the DTLZ1-DTLZ7 benchmark problems, SMPSO/CDLS, OMOPSO/CDLS and NMPSO/CDLS showed an obvious improvement on their original versions, as they receptively outperformed in 16, 19 and 11 out of 35 cases, while they only underperformed in 7, 6 and 5 cases. In addition, the *p*-value are respectively 0.0432, 0.0150 and 0.0193. When considering all the WFG1-WFG9 test problems, SMPSO/CDLS, OMOPSO/CDLS and NMPSO/CDLS can also obtain the best results on 17, 21 and 24 out of 45 cases, while their competitors only showed the best performance on 9, 3 and 0 out of 45 cases. The *p*-value for all the WFG test problems are 0.0570, 0.0000 and 0.0000, respectively. When considering the WFG41-WFG48 test problems, SMPSO/CDLS, OMOPSO/CDLS and NMPSO/CDLS also showed the superiority over their competitors respectively in 11, 37 and 13 out of 40 cases, while they only performed worse than their competitors in 5, 0 and 1 out of 40 cases, respectively. The *p*-values for all the WFG4X test problems are respectively 1.0000, 0.0000 and 0.0000.

Some similar conclusions can be learned from the results in Table 1. For solving MaOPs with most objectives, the improvement of SMPSO/CDLS, OMOPSO/CDLS and NMPSO/CDLS on their original algorithms is significant, as most of the *p*-values summarized in Table 1 are small or close to zero. Particularly, regarding the comparison between SMPSO/CDLS and SMPSO on MaOPs with 10 to 15 objectives, the CDLS strategy only improves the performance slightly. This may be mainly caused by the limited performance of the original PSO, where few particles with promising performance can be generated during the PSO search process, especially in a high-dimensional objective space (more than 10 objectives). That is to say, there are few suitable and promising particles in SMPSO which can be selected as the leader particles in the proposed CDLS strategy. Hence, the proposed CDLS strategy may not be very effective in such cases.

Furthermore, the total comparison results are summarized in the last row of Table 1. SMPSO/CDLS, OMOPSO/CDLS, and NMPSO/CDLS were superior to SMSPO, OMOPSO and NMSPO respectively in 44, 77 and 48 out of all 120 cases, and were only inferior to their competitors in 21, 9 and 6 out of 120 cases. Moreover, when considering the *p*-values for all the comparisons, they are respectively 0.0441, 0.0000 and 0.0000. Hence, the experimental comparison results listed in Table 1 have further verified the effectiveness of our proposed CDLS strategy, as the embedding of CDLS strategy into PSOs can bring significant improvements on tackling these MaOPs with various objectives.

4.6 A new variant PSO with the CDLS strategy

In this section, a new variant PSO is presented, namely PSO/CDLS. In addition to the proposed CDLS strategy, a novel space-division-based population update strategy is also designed in PSO/CDLS, which aims to provide more high-quality solutions for the PSO search process. By this way, the proposed population update strategy can be coordinated with our proposed CDLS strategy effectively.

Algorithm 4: *A*= **Archive Update** (*P, P'*)

```
Input: the current particle swarm $P'$, the old particle swarm $P$;
Output: the archive $A$;
$U=P' \cup P$;
$R$=Division ($U$)//Algorithm 5
$Remains=N$;
if $Remains > K$
    for $k$= 1 to $K$
        select the solution with the smallest sum value of objectives in $R_k$;
        add the selected solution to archive $A$ and remove it from $R_k$;
        $Remains=Remains-1$;
    end for
else
    select the remaining solutions from $|Remains|$ random subregions.
end if
```

The pseudo-code of the proposed population update strategy is given in **Algorithm 4**. At first, the current particle swarm *P'* and old particle swarm *P* are combined to formulate a union population *U* in line 1. Then, the division process is performed on *U* in line 2, which divides the objective space into *K* sub-regions by using the local density value Ld_i in (15). Please note that we set $K=\lceil N\times(M-1)/M\rceil$, where *N* and *M* respectively indicate the population size and the number of objectives. Note that a parameter sensitive experiment is conducted here to investigate the performance with different *K* values. Due to page limitations, the related empirical results with different sub-regions (*K*) are provided in Table S8 of supplementary file.

```
Algorithm 5: R=Division (U)
Input: the union population U;
Output: subregions R={R_1, R_2,..., R_K};
1:  set $S \leftarrow \varnothing$, $Q \leftarrow U$;
2:  set $R_k \leftarrow \varnothing$ ($k$=1,2,...,$K$);
3:  while $|Q| > K$
4:      for each solution in Q do
5:          $i \Leftarrow argmin\{Ld_1,\ldots, Ld_{|Q|}\}$;
6:          $Q \leftarrow Q \backslash \{x_i\}$;
7:          $S \leftarrow S \cup \{x_i\}$;
8:      end for
9:  end while
10: add $x_k \in Q$ into the subregion $R_k$ ($k$=1,2,...,$K$);
11: for each $x \in S$ do
12:     allocate $x$ to the closest subregion based on (5);
13: end for
```

The pseudo-code of subregion division is given in **Algorithm 5**. As shown in lines 3-9 of **Algorithm 5**, these solutions with smaller Ld_i are removed from the union population *Q* and added into *S*, and the rest *K* solutions in *Q* with uniform distribution are saved to construct *K* subregions in line 10. After that, an association procedure is used for the rest solutions in *S*. As shown in lines 11-13, each solution *x* for *S* is associated with the closest unique subregion based on the local density value Ld_i in (15).

After that, the proposed population update strategy is performed on the current particle swarm *P*. Specifically, when the numbers of the solution are larger than the numbers of subregion (i.e., *Remains*>*K*), the solutions are sequentially selected from each subregion according to their corresponding sum value of objectives in lines 5-9. Otherwise, the remaining solutions with the smallest sum value of objectives are selected from |*Remains*| random subregions in line 11. By this way, the proposed population update strategy can not only maintain the diversity by dividing the objective space into several uniform subregions, but also accelerate the convergence speed. In this way, this method can effectively coordinate with our proposed CDLS strategy.

4.6.1 Computational complexity analysis of PSO/CDLS

The complete process of PSO/CDLS is provided in **Algorithm 1**, consisting of four main procedures, i.e., the initialization process, the PSO search process, the evolutionary search process and population update process. Particularly, the initialization process is only run at the beginning of algorithm with the input population *P* with *N* solutions, which needs a time complexity of $O(mN)$, *m* is the number of objectives and *N* is the population size. For the PSO search process, the population *P* with *N* solutions is performed in **Algorithm 3** and **Algorithm 4**, which requires a total time complexity of $\mathrm{O}(mN)$. For evolutionary search process, as shown in line 12 of **Algorithm 1**, two reproduction operators, i.e., SBX and PM, are performed on the population *P* with a time complexity of $\mathrm{O}(mN)$. For the population update process, as shown in **Algorithm 5** and **Algorithm 6**, our designed space-division-based population update strategy is run on a union population *U* with 2*N* solutions, which needs a total time complexity of $\mathrm{O}(m|U||K|)$, where *U* and *K* are the size of union population (i.e., *U*=2*N*) and the numbers of sub-region (i.e., $K=\lceil N\times(m-1)/m\rceil$), respectively. Therefore, the overall time complexity of the proposed population update strategy is equal to $\mathrm{O}(mN^2)$. In summary, the worst time complexity of our proposed PSO/CDLS is $\mathrm{O}(mN^2)$ in each evolutionary iteration.

4.6.2 Comparison results on the adopted benchmark problems

In order to further study the performance of the proposed PSO/CDLS, three competitive PSOs (i.e., MPSO/D [26], AGPSO [31], and MaPSO [15]) and three EAs with promising performance for solving MaOPs (i.e., MaOEA/C [32], MaOEA/IT [33], and PAEA [34]) were used for comparison.

The HV compared results on WFG41-WFG48, WFG1-WFG9 and DTLZ1-DTLZ7 with 5 to 15 objectives are respectively summarized in Tables S9-S11. As learned from these results, PSO/CDLS also showed a superior performance over other competitors, as it obtained the best results on most cases. Specifically, PSO/CDLS obtained the best results in 25, 37 and 17 out of all 40 WFG4X test problems, 45 WFG test problems and 35 DTLZ test problems, respectively. Furthermore, the effectiveness of the proposed PSO/CDLS can be also verified according to the one-by-one comparison to each compared algorithm, because the proposed PSO/CDLS showed the better performance on solving most of the test problems adopted in our experiment. For the WFG4X test problems, PSO/CDLS showed superiority over MPSO/D, AGPSO, MaPSO, MaOEA/C, MaOEA/IT and PAEA respectively in 40, 23, 40, 17, 40 and 40 out of 40 cases, while it only underperformed its competitors in 0, 5, 0, 11, 0 and 0 out of 40 cases, respectively. Regarding the WFG test problems, PSO/CDLS outperformed MPSO/D, AGPSO, MaPSO, MaOEA/C, MaOEA/IT and PAEA respectively in 45, 30, 40, 32, 45 and 43 out of 45 case, while its competitors only showed advantages over PSO/CDLS in 0, 5, 4, 7, 0 and 1 out of 45 cases. For the DTLZ test problems, PSO/CDLS showed advantages in 26, 26, 22, 30, 35 and 25 out of 35 cases, respectively, while it underperformed MPSO/D, AGPSO, MaPSO, MaOEA/C, MaOEA/IT and PAEA respectively in 3, 2, 1, 3, 0 and 7 out of 35 cases. Hence, these experimental results listed in Tables S9-S11 have further validated the effectiveness of the proposed PSO/CDLS. In addition, due to page limitations, the IGD results obtained by each compared algorithm on all the used test problems are provided in the Tables S12-S14 of supplementary file, which also demonstrate the effectiveness of our proposed PSO/CDLS over its competitors in solving most adopted test problems.

To visually show and support the above discussions, some final solution sets with 15th best HV values from all 30 runs that are obtained by all the competitors are plotted in Figs. S1-S5 of supplementary file respectively for DTLZ2 with 5 objectives, WFG46 with 8 objectives, WFG5 with 10 objectives, WFG43 with 13 objectives and WFG47 with 15 objectives. Some conclusions can be learned from these figures, as they reflect the solution distribution for different test problems with 5 to 15 objectives. Compared with the competitors mentioned above, the final solution sets obtained by the proposed PSO/CDLS are distributed more evenly in these representative problems with different kinds of ***PF***s. Hence, the superiority of the proposed PSO/CDLS in solving these used test problems with different characteristics is further validated.

4.6.3 Further discussions and analysis on the overall performance of compared algorithms

To verify how well each compared algorithm performs overall, Friedman's test based on the software ***KEEL*** was also used here to rank all the competitors in all cases. Please note that a lower average performance rank indicates a better performance.

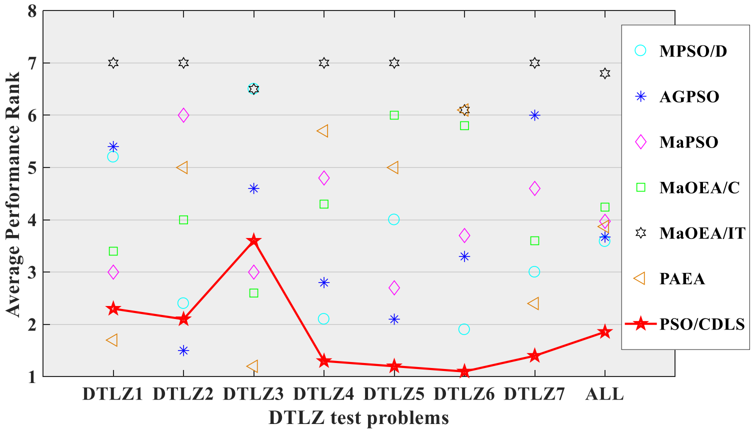

Fig. 4 Average performance rank in terms of all objectives dimensions for DTLZ test problems.

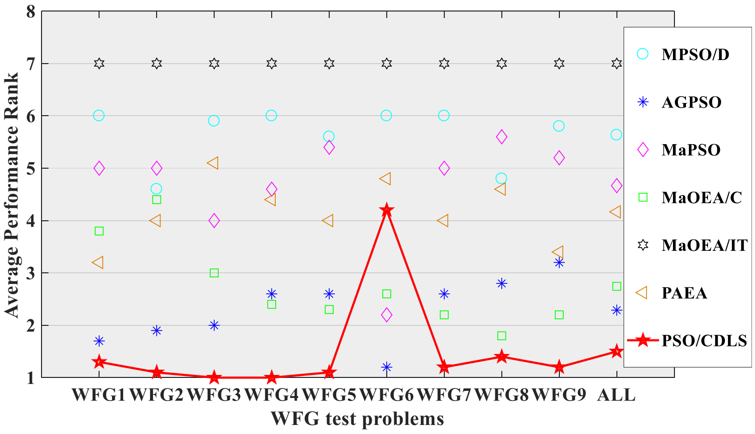

Fig. 5 Average performance rank in terms of all objectives for WFG test problems.

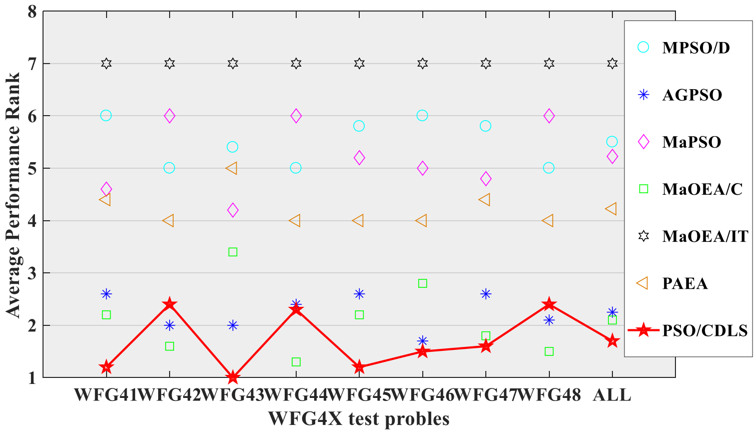

Fig. 6 Average performance rank in terms of all objectives for WFG4X test problems.

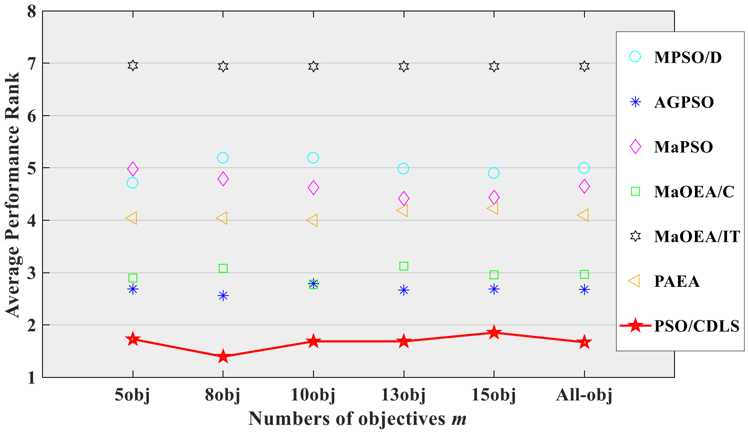

Fig. 7 Average performance rank over all test problems with different numbers of objectives.

Figs. 4-7 respectively provide the average performance ranks respectively on the DTLZ test problem, WFG test problems and WFG4X test problems with different numbers of objectives, including 5, 8, 10, 13, 15 and all numbers of objectives. To have an easy observation, a red line connects the average performance rank of the proposed PSO/CDLS. As shown in the last column of Figs. 4-6, it provides the average performance ranks in all the test cases. Specifically, the average performance rank in all DTLZ test problems with all objectives obtained by the proposed PSO/CDLS is 1.8571, followed by those of MPSO/D (3.5857), AGPSO (3.6714), PAEA (3.8714), MaPSO (3.9714), MaOEA/C (4.2429) and MaOEA/IT (6.8). The average performance rank in all the WFG test problems with all objectives for the proposed PSO/CDLS is 1.5, followed by those of AGPSO (2.2889), MaOEA/C (2.7444), PAEA (4.1667), MaPSO (4.6667), MPSO/D (5.6333) and MaOEA/IT (7). The average performance rank of the proposed PSO/CDLS in all the WFG4X test problems with all objectives is 1.7, followed by those of MaOEA/C (2.1), AGPSO (2.25), PAEA (4.225), MaPSO (5.225), MPSO/D (5.5) and MaOEA/IT (7). Hence, the effectiveness of the proposed PSO/CDLS is validated from those figures, as its average performance ranks on the three sets of test problems are much smaller for all cases when compared to its competitors.

Furthermore, Fig. 7 gives the average performance rank for different problems adopted in the experimental comparison, which has further validated the superiority of the proposed PSO/CDLS over its competitors, as the ranks of PSO/CDLS are the smallest in all numbers of objectives (i.e., 5 to 15 objectives). When considering the average performance rank as a whole that is listed in the last column of Fig. 7, the average performance rank of PSO/CDLS is 1.6708, followed by those of AGPSO (2.6792), MaOEA/C (2.9667), PAEA (4.1), MaPSO (4.65), MPSO/D (4.9917) and MaOEA/IT (6.9417). To conclude, the superiority of PSO/CDLS on most cases is further verified by observing Figs. 4-7.

5. Conclusions and future work

This paper presented a convergence and diversity guided leader selection (CDLS) strategy for PSOs, in which the leader selection process is guided based on the status of convergence and diversity. In this strategy, suitable leader particles will be adaptively selected for each particle in each of four statuses. Specifically, $gbest_i^C$ is used to guide the particle with a poor convergence performance, which is expected to accelerate the convergence speed. Conversely, $gbest_i^D$ is assigned for the particle with a good performance on convergence, which aims to improve its diversity performance. $lbest_i^C$ is selected for the particle located at a crowded area, which is also expected to provide a strong convergence pressure. On the other hand, $lbest_i^D$ is selected for the particle located at a sparse region, aiming to improve the distribution of the whole swarm. When CDLS is embedded into three competitive PSOs, such as SMPSO, OMOPSO, and NMPSO, the experimental results validated that their performance would be improved significantly when solving the WFG, DTLZ and WFG4X test problems with 5 to 15 objectives. Moreover, a novel PSO algorithm combining the proposed CDLS (named PSO/CDLS) was also presented in this paper. When compared to three competitive PSOs (MPSO/D, AGPSO and MaPSO) and three competitive EAs (MaOEA/C, MaOEA/IT and PAEA), PSO/CDLS also showed an obvious superiority in tackling most of the test problems adopted.

In our future work, the CDLS strategy will be further studied for solving the problems with a high-dimensional decision variable space, namely large-scale optimization problems [61]-[62],. Moreover, the performance of particle swarm optimizer combining some transfer learning methods for solving some multi/many-tasking problems [63]-[64] will be also considered in our future work.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants 61876110 and 61836005, the Shenzhen Scientific Research and Development Funding Program under Grant JCYJ20190808164211203, the Guangdong "Pearl River Talent Recruitment Program" under Grant 2019ZT08X603, and Shenzhen Science and Technology Innovation Commission (R2020A045). Prof. Coello Coello gratefully acknowledges support from CONACyT grant no. 2016-01-1920 (Investigación en Fronteras de la Ciencia 2016). He was also partially supported by the Basque Government through the BERC 2022-2025 program and by Spanish Ministry of Economy and Competitiveness MINECO: BCAM Severo Ochoa excellence accreditation SEV-2017-0718.

Reference

[1] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation, 6 (2) (2002) 182-197.

[2] K. Miettinen, Nonlinear multiobjective optimization. Boston, Ma, USA: Kluwer Academic, 1999.

[3] G.S. Li, T. Zhou, A multi-objective particle swarm optimizer based on reference point for multimodal multi-objective optimization, Engineering Applications of Artificial Intelligence, 107 (2022) 104523.

[4] C. He, S. Huang, R. Cheng, K. C. Tan and Y. Jin, Evolutionary multiobjective optimization driven by generative adversarial networks (GANs), IEEE Transactions on Cybernetics, 51 (6) (2021) 3129-3142.

[5] S. Kahloul, D. Zouache, B. Brahmi and A. Got, A multi-external archive-guided henry gas solubility optimization algorithm for solving multi-objective optimization problems, Engineering Applications of Artificial Intelligence, 109 (2022) 104588.

[6] X.W. Zhang, H. Liu and L.P. Tu, A modified particle swarm optimization for multimodal multi-objective optimization, Engineering Applications of Artificial Intelligence, 95 (2020) 103905.

[7] L. Wang, Q. Zhang, A. Zhou, M. Gong and L. Jiao, Constrained subproblems in a decomposition-based multiobjective evolutionary algorithm, IEEE Transactions on Evolutionary Computation, 20 (3) (2016) 475-480.

[8] L. Li, Q. Lin and Z. Ming, Multi-objective optimization using self-organizing decomposition and its application to crashworthiness design, Applied Soft Computing, 101 (2021) 107002.

[9] W.F. Li, L.J. He, and Y.L. Cao, Many-objective evolutionary algorithm with reference point-based fuzzy correlation entropy for energy-efficient job shop scheduling with limited workers, IEEE Transactions on Cybernetics, 2021, doi: 10.1109/TCYB.2021.3069184.

[10] Y.L. Cao, H. Zhang, W.F. Li, M.C. Zhou, Y. Zhang, W.A. Chaovalitwongse, Comprehensive learning particle swarm optimization algorithm with local search for multimodal functions, IEEE Transactions on Evolutionary Computation, 23 (4) (2019) 718-731.

[11] A. J. Nebro, J. J. Durillo, J. Garcia-Nieto, C. A. Coello Coello, F. Luna and E. Alba, SMPSO: A new PSO-based metaheuristic for multi-objective optimization, 2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making(MCDM), Nashville, TN, 2009, pp. 66-73, doi: 10.1109/MCDM.2009.4938830.

[12] Z. Zhan, J. Li, J. Cao, J. Zhang, H. S. Chung and Y. Shi, Multiple populations for multiple objectives: a coevolutionary technique for solving multiobjective optimization problems, IEEE Transactions on Cybernetics, 43 (2) (2013) 445-463.

[13] Sierra M.R., Coello Coello C.A, Improving PSO-based multi-objective optimization using crowding, mutation and ∈-Dominance. In: Coello Coello C.A., Hernández Aguirre A., Zitzler E. (eds) Evolutionary Multi-Criterion Optimization. EMO 2005. Lecture Notes in Computer Science, vol 3410. Springer, Berlin, Heidelberg.

[14] Q. Lin, S. Liu, C. Tang, R. Song, J. Chen and C. A. Coello Coello, Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems, IEEE Transactions on Evolutionary Computation, 22 (1) (2018) 32-46.

[15] Y. Xiang, Y. Zhou, Z. Chen and J. Zhang, A many-objective particle swarm optimizer with leaders selected from historical solutions by using scalar projections, IEEE Transactions on Cybernetics, 50 (5) (2020) 2209-2222.

[16] H. Han, W. Lu, L. Zhang and J. Qiao, Adaptive gradient multiobjective particle swarm optimization, IEEE Transactions on Cybernetics, 48 (11) (2018) 3067-3079.

[17] W. Hu and G. G. Yen, Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system, IEEE Transactions on Evolutionary Computation, 19 (1) (2015) 1-18.

[18] B. Wu, W. Hu, J. Hu and G. G. Yen, Adaptive multiobjective particle swarm optimization based on evolutionary state estimation, IEEE Transactions on Cybernetics, 51 (7) (2021) 3738-3751.

[19] H. Han, W. Lu and J. Qiao, An adaptive multiobjective particle swarm optimization based on multiple adaptive methods, IEEE Transactions on Cybernetics, 47 (9) (2017) 2754-2767.

[20] W. Hu, G. G. Yen and G. Luo, Many-objective particle swarm optimization using two-stage strategy and parallel cell coordinate system, IEEE Transactions on Cybernetics, 47 (6) (2017) 1446-1459.

[21] L. Li, L. Chang, T. Gu, W. Sheng and W. Wang, On the norm of dominant difference for many-objective particle swarm optimization, IEEE Transactions on Cybernetics, 51 (4) (2021) 2055-2067.

[22] W. Peng and Q. Zhang, A decomposition-based multi-objective particle swarm optimization algorithm for continuous optimization problems, 2008 IEEE International Conference on Granular Computing, Hangzhou, 2008, pp. 534-537.

[23] Saúl Zapotecas Martínez and Carlos A. Coello Coello, A multi-objective particle swarm optimizer based on decomposition. In Proceedings of the 13th annual conference on Genetic and evolutionary computation (GECCO). Association for Computing Machinery, New York, NY, USA, pp.69–76. 2011.

[24] N.A. Moubayed, A. Petrovski and J. McCall, D^{2}MOPSO: MOPSO based on decomposition and dominance with archiving using crowding distance in objective and solutions spaces, Evolutionary Computation, 22 (1) (2015) 47-77.

[25] W. Liu, Z. Wang, Y. Yuan, N. Zeng, K. Hone and X. Liu, A novel sigmoid-function-based adaptive weighted particle swarm optimizer, IEEE Transactions on Cybernetics, 51 (2) (2021) 1085-1093.

[26] C. Dai, Y. Wang and M. Ye, A new multi-objective particle swarm optimization algorithm based on decomposition, Information Sciences, 325 (2015) 541-557.

[27] X. Yu, W. Chen, T. Gu, H. Zhang, H. Yuan, S. Kwong and J. Zhang, Set-based discrete particle swarm optimization based on decomposition for permutation-based multiobjective combinatorial optimization problems, IEEE Transactions on Cybernetics, 48 (7) (2018) 2139-2153.

[28] S. Huband, L. Barone, R. While, and P. Hingston, A scalable multi-objective test problem toolkit, in Proc. 3rd Conference Evolutionary Multi Criterion Optimization, Guanajuato, Mexico, 2005, pp. 280–295.

[29] R. Wang, R.C. Purshouse and P.J. Fleming, Preference-inspired co-evolutionary algorithms using weight vectors, European Journal of Operational Research, 243 (2) (2015) 423-441.

[30] J. Kennedy and R. Eberhart, Particle swarm optimization, Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, 1995, pp. 1942-1948 vol.4, doi: 10.1109/ICNN.1995.488968.

[31] F. Chen, S. Wu, F. Liu, J. Ji and Q. Lin, A novel angular-guided particle swarm optimizer for many-objective optimization problems, Complexity, 2020 (2020) 6238206.

[32] Q. Lin, S. Liu, K. Wong, M. Gong, Carlos A. Coello Coello, J. Chen and J. Zhang, A clustering-based evolutionary algorithm for many-objective optimization problems, IEEE Transactions on Evolutionary Computation, 23 (3) (2019) 391-405.

[33] Sun, B. Xue, M. Zhang and G. G. Yen, A new two-stage evolutionary algorithm for many-objective optimization, IEEE Transactions on Evolutionary Computation, 23 (5) (2019) 748-761.

[34] Y. Zhou, Y. Xiang, Z. Chen, J. He and J. Wang, A scalar projection and angle-based evolutionary algorithm for many-objective optimization problems, IEEE Transactions on Cybernetics, 49 (6) (2019) 2073-2084.

[35] Y. Yuan, H. Xu, B. Wang, B. Zhang and X. Yao, Balancing convergence and diversity in decomposition-based many-objective optimizers, IEEE Transactions on Evolutionary Computation, 20 (2) (2016) 180-198.

[36] Q. Zhu, Q. Lin, W. Chen, K. Wong, Carlos A. Coello Coello, J. Li and J. Chen, An external archive-guided multiobjective particle swarm optimization algorithm, IEEE Transactions on Cybernetics, 47 (9) (2017) 2794-2808.

[37] S. Das and P. N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Transactions on Evolutionary Computation, 15 (1) (2011) 4-31.

[38] S. Brindha and S. Miruna Joe Amali, A robust and adaptive fuzzy logic based differential evolution algorithm using population diversity tuning for multi-objective optimization, Engineering Applications of Artificial Intelligence, 102 (2021) 104240.

[39] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, Scalable test problems for evolutionary multiobjective optimization, evolutionary multiobjective optimization (advanced information and knowledge processing), A. Abraham, L. Jain, and R. Goldberg, Eds. London, U.K.: Springer, 2005, pp. 105–145.

[40] X. Zhang, X. Zheng, R. Cheng, J. Qiu and Y. Jin, A competitive mechanism based multi-objective particle swarm optimizer with fast convergence, Information Sciences, 427 (2018) 63-76.

[41] Q. Lin, J. Li, Z. Du, J. Chen and Z. Ming, A novel multi-objective particle swarm optimization with multiple search strategies, European Journal of Operational Research, 247 (3) (2015) 732-744.

[42] Carlo R. Raquel and Prospero C. Naval, An effective use of crowding distance in multiobjective particle swarm optimization. In Proceedings of the 7th annual conference on Genetic and evolutionary computation (GECCO). Association for Computing Machinery, New York, NY, USA, pp.257–264, 2005.

[43] K. Deb and H. Jain, An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: solving problems with box constraints, IEEE Transactions on Evolutionary Computation, 18 (4) (2014) 577–601.

[44] P.A.N. Bosman and D. Thierens, The balance between proximity and diversity in multiobjective evolutionary algorithms, IEEE Transactions on Evolutionary Computation, 7 (2) (2003) 174–188.

[45] Y. Xiang, Y. R. Zhou, and M. Q. Li, A vector angle based evolutionary algorithm for unconstrained many-objective optimization, IEEE Transactions on Evolutionary Computation, 21 (1) (2017) 131–152.

[46] X. Li and X. Yao, Cooperatively coevolving particle swarms for large scale optimization, IEEE Transactions on Evolutionary Computation, 16 (2) (2012) 210-224.

[47] K. Li, K. Deb, Q. Zhang and S. Kwong, An evolutionary many-objective optimization algorithm based on dominance and decomposition, IEEE Transactions on Evolutionary Computation, 19 (5) (2015) 694-716.

[48] Y. Yang, Y.R. Zhou, X.W. Yang and H. Huang, A many-objective evolutionary algorithm with pareto-adaptive reference points, IEEE Transactions on Evolutionary Computation, 24 (1) (2020) 99–113.

[49] S.B. Liu, Q.Y. Yu, Q.Z. Lin and K.C. Tan, An adaptive clustering-based evolutionary algorithm for many-objective optimization problems, Information Sciences, 537 (2020) 261-283.

[50] K. Li, Q. Zhang, S. Kwong, M. Li and R. Wang, Stable matching-based selection in evolutionary multiobjective optimization, IEEE Transactions on Evolutionary Computation, 18 (6) (2014) 909-923.

[51] L.J. Li, Q.Z. Lin, K. Li and Z. Ming, Vertical distance-based clonal selection mechanism for the multiobjective immune algorithm, Swarm and Evolutionary Computation, 63 (2021) 100886.

[52] T. Zhou, Z.B. Hu, Q. Zhou and S.X. Yuan, A novel grey prediction evolution algorithm for multimodal multiobjective optimization, Engineering Applications of Artificial Intelligence, 100 (2021) 104173.

[53] X. Zhang, Y. Tian and Y. Jin, A knee point-driven evolutionary algorithm for many-objective optimization, IEEE Transactions on Evolutionary Computation, 19 (6) (2015) 761-776.

[54] Y. Tian, R Cheng and X Zhang, A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization, IEEE Transactions on Evolutionary Computation, 23 (2) (2018) 331-345.

[55] J. Alcala-Fdez et al., KEEL: A software tool to assess evolutionary algorithms for data mining problems, Soft Computing, 13 (2009) 307–318.

[56] J.J. Durillo and A.J. Nebro, jMetal: A Java framework for multi-objective optimization, Advances in Engineering Software, 42 (10) (2011) 760-771.

[57] Y. Tian, R. Cheng, X. Zhang and Y. Jin, PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum], IEEE Computational Intelligence Magazine, 12 (4) (2017) 73-87.

[58] Y. Tian, X.S. Xiang, X.Y. Zhang, R. Cheng, and Y.C. Jin, Sampling reference points on the Pareto fronts of benchmark multi-objective optimization problems, 2018 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2018.

[59] K. Deb and R. B. Agrawal, Simulated binary crossover for continuous search space, Complex Systems, 9 (3) (1994) 115–148.

[60] H. Li, Q.F. Zhang, Multi-objective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, IEEE Transactions on Evolutionary Computation, 13 (2) (2009) 284–302.

[61] R. Moeini, M. Soltani-nezhad, and M. Daei, Constrained gravitation search algorithm for large scale reservoir operation optimization problem, Engineering Applications of Artificial Intelligence, 61 (2017) pp. 222-233.

[62] S. Qi, J. Zou, S.X. Yang, J.H. Zheng, A level-based multi-strategy learning swarm optimizer for large-scale multi-objective optimization, Swarm and Evolutionary Computation, 73 (2022) 101100.

[63] J. Lin, H. -L. Liu, K. C. Tan and F. Gu, An effective knowledge transfer approach for multiobjective multitasking optimization, IEEE Transactions on Cybernetics, 51 (6) (2021) 3238-3248.

[64] Y.Q. Cai, D.M. Peng, P.Z. Liu, and J.M. Guo, Evolutionary multi-task optimization with hybrid knowledge transfer strategy, Information Sciences, 580 (2021) pp. 874-896.