A Self-organizing Weighted Optimization based Framework for Large-scale Multi-objective Optimization

Yongfeng Li[1], Lingjie Li[1], Qiuzhen Lin[1]*, Ka-Chun Wong[2], Zhong Ming[1], Carlos A. Coello Coello[3,4]

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

[2] *Department of Computer Science, City University of Hong Kong, Hong Kong SAR*

[3]*CINVESTAV-IPN, Department of Computer Science, Mexico, D.F., 07360, Mexico*

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

Abstract:

The solving of large-scale multi-objective optimization problem (LSMOP) has become a hot research topic in evolutionary computation. To better solve this problem, this paper proposes a self-organizing weighted optimization based framework, denoted S-WOF, for addressing LSMOPs. Compared to the original framework, there are two main improvements in our work. Firstly, S-WOF simplifies the evolutionary stage into one stage, in which the evaluating numbers of weighted based optimization and normal optimization approaches are adaptively adjusted based on the current evolutionary state. Specifically, regarding the evaluating number for weighted based optimization (i.e., t_1), it is larger when the population is in the exploitation state, which aims to accelerate the convergence speed, while t_1 is diminishing when the population is switching to the exploration state, in which more attentions are put on the diversity maintenance. On the other hand, regarding the evaluating number for original optimization (i.e., t_2), which shows an opposite trend to t_1, it is small during the exploitation stage but gradually increases later. In this way, a dynamic trade-off between convergence and diversity is achieved in S-WOF. Secondly, to further improve the search ability in the large-scale decision space, an efficient competitive swarm optimizer (CSO) is implemented in S-WOF, which shows efficiency for solving LSMOPs. Finally, the experimental results have validated the superiority of S-WOF over several state-of-the-art large-scale evolutionary algorithms.

Keywords: Large-scale optimization; Weighted optimization; Competitive swarm optimizer (CSO)

1. Introduction

In recent years, multi-objective optimization problems (MOPs), consisting of two or more conflicting objectives, are commonly seen in many real-world engineering applications [1-3], such as power allocation [4], dynamic resource strategy [5], shop scheduling [6], chemical engineering [7] and crashworthiness design [8]. In general, there is no single solution that can optimize all the conflicting objectives simultaneously. Therefore, a set of tradeoff solutions, namely Pareto-optimal set (***PS***), is expected to obtain when solving each MOP and its mapping to the objective space is termed Pareto-optimal (***PF***). The main goal of solving MOPs is to find a set of solutions that can distribute as closely as possible and evenly around the true ***PF***. Over the several decades, multi-objective evolutionary algorithms (MOEAs) have been well studied and shown advantages for solving MOPs. Most existing MOEAs can be broadly grouped into three categories, including Pareto-based MOEAs [9-13], decomposition-based MOEAs [14-18] and indicator-based MOEAs [19-23].

For the Pareto-based MOEAs, they use the Pareto dominance relationship to select solutions. For instance, NSGA-II [9] was presented based on the Pareto dominance relationship, which uses the nondominated sorting method to decide the front number of the population and adopts the crowding distance to maintain the diversity of the population. SPEA [10] combined several features of evolutionary algorithm in a unique manner, i.e., storing nondominated solutions in an external archive, evaluating the fitness of each solution dependent on the number of external nondominated solutions, maintaining population diversity based on the Pareto dominance relationship, and incorporating a clustering method. ISPEA [11] was proposed to restrain degeneracy of the evolutionary process by the immune operator, which includes vaccine extraction, vaccination, and immune selection in turn. SPEA2 [12] not only used the fitness assignment method to consider both the dominated number and dominate number of each individual, but also applied the truncation method to guarantee the preservation of boundary solutions. PDTEA [13] proposed a novel mating selection strategy to select two solutions with good diversity and convergence by a decomposition-based method, and adopted the novel environmental selection to maintain the diversity by using a modified truncation method. Moreover, an efficient dynamic response mechanism was also presented in PDTEA, aiming to produce more promising solutions with good convergence and diversity when the environment changes.

The decomposition-based MOEAs decompose the MOP into several subproblems by decomposition approaches. Particularly, MOEA/D [14] is one of the representative algorithms of decomposition-based MOEAs. However, MOEA/D was deteriorated when solving some problems with irregular ***PF***s. To address this issue, many decomposition-based methods were presented recently, which showed promising performance. For example, DMaOEA-εC [15] not only adopted a maintenance mechanism for boundary points and a distance-based replacement strategy to keep the diversity of the population, but also applied a two-side update rule to accelerate the convergence of the population. IBMOEA/D [16] proposed a decomposition-based strategy to evolve the population, in which each individual stands for a subproblem. Then a binary quality indication-based selection was applied to save the external population, which integrates preference information into search. Lin *et.al* proposed a novel adaptive operator selection (B-AOS) strategy [17] for decomposition-based MOEAs, which applies two operator pools to focus on exploitation and exploration respectively, where two different DE operators are adaptively adjusted according to the evolutionary state of the population. Moreover, FDEA-I and FDEA-II [18] were proposed based on the fractional dominance relation and the improved objective space decomposition strategy. Particularly, the first one is proposed to retain some solutions with promising performance, while the second one is presented to improve the ability of diversity maintenance, respectively.

For the third category, namely indicator-based MOEAs, they select the solutions by some indicators. For example, AR-MOEA [19] designed a distance based indicator, in which the reference vectors are dynamically adjusted according to the corresponding indicator contributions of the solutions in an external archive. hpaEA [20] proposed the hyperplane that is formed by the neighboring solutions of nondominated solutions. In addition, an efficient environmental selection strategy based on the hyperplane produced by neighboring solutions of nondominated solutions was also implemented in hpaEA, which aims to further distinguish nondominated solutions. Besides that, R2HCA-EMOA [21] adopted an R2 indicator variant to approximate the hypervolume contribution. EIMA [22] proposed an adaptive reference point, which is adaptively adjusted by the calculation of indicators. LIBEA [23] proposed a Lebesgue indicator-based algorithm, which adopts the regularity property of continuous multiobjective problems (MOPs) to solve MOPs with difficult properties.

Although the above MOEAs have shown their advantages in solving traditional problems with low dimensional space, they behaved poorly when solving large-scale optimization problems (LSMOPs) consisting of a large-scale number of decision variables, due to the low search efficiency of operators (e.g., simulated binary crossover (SBX) [24], polynomial mutation (PM) [25], and differential evolution (DE) [26]). Particularly, a self-adaptive DE operator, termed njDE, was proposed in [27] to improve the global search ability. More recently, many MOEAs were extended to solve LSMOPs. These algorithms can be roughly divided into two main categories, including cooperative coevolutionary algorithms (CCEAs) [28-32] with the cooperative idea to optimize LSMOPs and competitive swarm optimizers (CSOs) [33-37] that are considered as an improved variant based on particle swarm optimizer (PSO) as most existing PSOs [38-41] converged prematurely when solving LSMOPs with a large-scale decision space. Although CCEAs and CSOs performed well on solving LSMOPs, the problems involving premature convergence [42] and a huge number of evaluations still existed. To ameliorate these issues, a self-organizing weighted optimization based framework, namely S-WOF, is proposed in this paper. Specifically, the main contributions of this paper are summarized as follows:

(1) Compared to the original framework in [42], the optimization of transformed problem and original problem in the proposed S-WOF are integrated into one stage, which is more efficient. In addition, a self-organizing mechanism is designed in S-WOF, which can adaptively adjust the evaluating numbers for weighted based optimization (i.e., t_1) and original optimization (i.e., t_2) based on the current evolutionary information. In this way, a dynamic trade-off between convergence and diversity is achieved in S-WOF.

(2) In order to improve the search ability in a huge search space, an efficient CSO variant with a strong exploration ability is also implemented in S-WOF, which applies a novel particle updating strategy to enhance the search efficiency.

(3) To verify the effectiveness of S-WOF, nine test LSMOPs with up to 3 objectives and 1000 decision variables are adopted in our experiments, which have complicated Pareto-optimal sets and fronts. Four competitive large-scale MOEAs, i.e., WOF-NSGA-II [43], LMOCSO [36], DGEA [44] and LMEA [45], are adopted for performance comparisons. The experimental results have validated the advantages of our proposed S-WOF over these competitive large-scale MOEAs on most test problems adopted.

The rest of this paper is organized as follows. **Section 2** provides some background information, including the basic introduction of MOPs and a brief review of large-scale MOEAs. **Section 3** gives the details of our proposed S-WOF. **Section 4** provides the experimental comparisons between S-WOF and other compared large-scale MOEAs. Finally, **Section 5** gives the conclusions and some future work.

2. Background

Many real world problems need to optimize multiple conflicting objectives simultaneously, which are often called multi-objective optimization problems (MOPs). Without loss of generality, a minimum MOP can be formulated as follows:

$$\min_{x\in\Omega} f(x) = (f_1(x), f_2(x), ..., f_m(x))^T \tag{1}$$

where $x = (x_1, x_2, \ldots, x_n) \in \Omega$ is a decision vector with n dimensions, Ω is the feasible region of decision space, the mapping function $f: \Omega \rightarrow R^m$ defines m objectives and R^m is the objective space.

With the increase of decision variable dimensions, the above problem becomes an LSMOP. In recent years, many MOEAs were proposed to solve LSMOPs. These MOEAs can be categorized into two groups, including cooperative coevolutionary algorithms (CCEAs) and competitive swarm optimizers (CSOs), where some related studies on CCEAs and CSOs for solving LMOPs are introduced in **Section 2.1** and **Section 2.2**, respectively.

2.1 Cooperative coevolutionary algorithms

Since Potter *et al.* proposed the cooperative coevolution (CC) framework, which applies the divide-and-conquer technique to decompose LSMOPs into smaller subproblems and then optimizes these subproblems independently. After that, various MOEAs were presented under the CC framework to solve LSMOPs. For example, Li *et al.* proposed CCPSO2 [28], which adopts Cauchy and Gaussian distributions to update the position of a particle and determines the number of decision variables of a group. In addition, a new decomposition strategy called differential grouping (DG) [31] was proposed by Li *et al.*, which aims to improve the distributions of decision variables in CC algorithms. Apart from CC framework, Zille *et al.* proposed a weighted optimization based framework [43], denoted as WOF, which divides the decision variables into several groups and applies the same weight value to the decision variables in the same group. Ma *et al*. proposed MOEA/DVA [46] based on variable analysis, which optimizes the distance variables at first before focusing on the position variables. In this way, MOEA/DVA concentrates on the convergence in the early stage and considers the diversity in the later stage. LMEA [45] was proposed by Zhang *et al.*, which adopts a clustering method to classify decision variables into distance-related variables and position-related variables, which are then optimized independently. LSMOF [47] was presented by reformulating the original LSMOPs into a low-dimensional single-objective problem, in which the decision place is reconstructed by using the weight variables in the reformulated problem and the objective space is reduced by using an indicator function. Although these CCEAs have shown a good performance for solving LSMOPs, they may encounter two main challenges. Regarding the first one, these CCEAs seriously depend on the used decomposition strategy, which usually determines the measure of the performance. Therefore, the selection of a good decomposition method plays an important role in LSMOPs. Regarding the second one, detecting the internship among variables tends to consume a great number of function evaluations, while finding a good decomposer also requires a great deal of function evaluations. Hence, most of them suffer from high computational complexity.

2.2 Competitive swarm optimizers

Recently, Cheng *et al.* proposed an improved PSO variant, called CSO [48], which introduces the pairwise competitive within a swarm. Different from traditional PSO that applies the global best and personal best particles to guide the evolution of swarm, CSO introduces the pairwise competitive mechanism to divide the particles into two groups based on the fitness value of each particle, including winner particle group with better fitness values and loser particle group with lower fitness values. Then, the position and velocity of loser particles are updated by learning from their corresponding winner particles, formulated as follows:

$$\begin{aligned} V_l(t+1) &= r_1(t)V_l(t) + r_2(t)(X_w(t) - X_l(t)) + \psi r_3(t)(\overline{X}(t) - X_l(t)) \\ X_l(t+1) &= X_w(t) + V_l(t+1) \end{aligned} \tag{1}$$

where $V_l(t)$ and $X_l(t)$ are the velocity and position of the loser in the t-th generation, $r_i(t)$ (i = 1, 2, 3) is the random variable within [0,1] in the t-th generation, $X_w(t)$ is the position of the winner in the t-th generation and ψ is the control parameter of $\overline{X}(t)$.

More recently, some CSO variants were proposed for solving LSMOPs, which have shown promising performance due to the efficient search ability of CSO. For example, two learning processes were proposed in TPLSO [33], including mass learning and elite learning. Specifically, particles in mass learning were divided into several groups. In each group, the particles competed with each other, and then the winner would be saved and enter into elite learning. On other hands, particles in swarm were sorted according to their fitness value during the process of elite learning. In this way, each particle will learn from two particles that are better than it is. In SPLSO [34], two different learning mechanisms were proposed during the entire process. Specifically, segment-based learning strategy divided the decision variables into several segments and then each segment of a particle learned from one exemplar. On other hands, the segment-based predominant learning mechanism divided particles into good ones and poor ones and then the poor ones can be further optimized by learning from good ones. In LLSO [35], the particles were divided into four levels according to their fitness value. Afterward, particles in each level learned from those particles that were in lower levels. In LMOCSO [36], the fitness value of particles in the swarm was first calculated by shift-based density estimation (SDE) [49]. Then, two particles were randomly selected to compete with each other. After that, the worse particle would learn from the better particle. Different from most traditional CSOs, two-thirds of the population were updated according to a tri-competitive criterion in MCSO [37], where the particles are divided into several groups and each group includes three particles. Then, the second-best particle (X_{L1}) learned from the best particle (X_w) and the mean particle ($\overline{X}$). The worst particle learned as same as X_{L1}. However, even though the existing CSOs have been proposed for solving LSMOPs and shown strong exploration ability, they still fall into local optima and converge prematurely to some extent.

To conclude, as introduced above, most existing CCEAs require a great deal of function evaluations, while most traditional CSOs may encounter the problems of premature convergence. Inspired from that, this paper proposes a self-organizing weighted optimization based framework, namely S-WOF, consisting of a self-organizing mechanism and an efficient CSO variant, which can achieve a good balance between the convergence and diversity, as well as shows a strong search ability when solving LSMOPs in a huge search space. The details of our proposed S-WOF are introduced in next section.

3. The proposed algorithm

In this section, the details of the proposed S-WOF are described. Firstly, the complete framework of the proposed S-WOF is introduced in **Section 3.1**. Then, its two main components, including the proposed efficient CSO variant and the proposed self-organizing mechanism, are introduced in **Section 3.2** and **Section 3.3** successively.

3.1. The complete framework of the proposed S-WOF

```
Algorithm 1: The complete framework of S-WOF
Input: MaxFes
Output: P
1    P ← randomly initialize population;
2    set fes=|P|, t1, t2;
3    while fes<MaxFes do
4        W ←randomly initialize weight vectors;
5        Pw ←apply W on P ;
6        P' ←WeightOptimization_by_CSO (Pw ,t1); //Algorithm 2
8        Offspring ←NormalOptimization_by_CSO (P' ,t2); //Algorithm 2
9        t1, t2←Self-organizing Mechanism (P', P ); //Algorithm 3
10       fes= fes+ t1+ t2;
11       P = Offspring;
12   end while
```

The pseudo-code of the proposed S-WOF is given in **Algorithm 1**. At the beginning, the initialization procedure is performed in lines 1-2, in which *fes*, t_1 and t_2 are the current evaluating numbers, the evaluating numbers for weighted optimization method and normal optimization method, respectively. Then, S-WOF enters the main loop as shown in lines 3-12, which will terminate until *fes* reaches the predefined maximum number of evaluations (i.e., *MaxFes*). In each iteration of the main loop, the weight vector set W are randomly initialized at first in line 4. Then, as shown in line 5, the new population P_w is generated by applying W on P. After that, the weighted optimization method is performed on P_w with t_1 evaluating numbers and then the new population P' is generated. Please note that a novel efficient CSO variant is designed as evolutionary search operator, which is introduced in **Algorithm 2 of Section 3.2**. After the weighted optimization method is completed, the normal optimization method is further performed on P' with t_2 evaluating numbers. As shown in line 8, *Offspring* is obtained by using the efficient CSO operator. Afterwards, the evaluating numbers for both weighted based optimization method (i.e., t_1) and normal optimization method (i.e., t_2) are updated by the proposed self-organizing mechanism in line 9. Note that the detail of the proposed self-organizing mechanism is introduced in **Algorithm 3 of Section 3.3.** Finally, the final solutions in P are reported as the approximation ***PS***.

3.2. An efficient competitive swarm optimizer in S-WOF

Algorithm 2: $P' \leftarrow$WeightOptimization_by_***CSO*** (P_w, t_1);
Input: P_w, t_1
Output: P'
1 *Offspring* $\leftarrow \emptyset$, $f_{SDE} \leftarrow 0$, $P' \leftarrow P_w$, $P \leftarrow \emptyset$;
2 **while** *fes* $< t_1$ **do**
3 $\quad P \leftarrow P'$;
4 $\quad f_{SDE} \leftarrow$ compute the fitness value for each particle in P using Eq. (2);
5 $\quad$**for** each particle in P **do**
6 $\qquad (X_w, X_{L1}, X_{L2}) \leftarrow$ randomly select three particles from P;
7 $\qquad P \leftarrow P \setminus (X_w, X_{L1}, X_{L2})$;
8 $\qquad (X_w, X_{L1}, X_{L2}) \leftarrow$ swap positions based on f_{SDE} in Eq.(2);
9 $\qquad$update X_{L1} and X_{L2} by using Eq. (3);
10 $\qquad$mutate X_{L1}, X_{L2} and X_w;
11 $\qquad$*Offspring* $\leftarrow (X_w, X_{L1}, X_{L2}) \cup$ *Offspring*;
12 $\quad$**end for**
13 $\quad P' \leftarrow$EnvironmentalSelection(*Offspring* $\cup P'$);
14 **end while**

The pseudo-code of the proposed Weightedoptimization_***CSO*** is given in **Algorithm 2**, with the inputs: the current population P_w and the evaluating numbers of weighted optimization t_1. At the beginning, the child population *Offspring*, the fitness value f_{SDE} for each individual and the two new population P' and P are initialized in line 1. Then, the proposed Weightedoptimization_***CSO*** steps into the main loop of the evolutionary process, which will be terminated when the evaluations of the given problem (*fes*) reach t_1. In the main loop, as shown in line 4, the f_{SDE} value for each particle in the swarm is calculated as follows:

$$f_{\mathrm{SDE}} = \min_{q \in P \setminus \{p\}} \sqrt{\sum_{i=1}^{M} (\max\{0, f_i(\vec{q}) - f_i(\vec{p})\})^2} \tag{2}$$

where $f_i(\vec{p})$ denotes the *i*th objective value of solution *p*. After calculating the fitness value of each particle in P, the CSO-based update is performed in lines 5-12. Specifically, as shown in lines 6-8, three particles are randomly selected from P firstly and then one winner particle (i.e., X_w), and two loser particles (i.e., X_{L1} and X_{L2}) are determined in line 8 according to their corresponding f_{SDE} value. After that, the position and velocity of X_{L1} and X_{L2} are updated by learning from X_w in line 9. The details of the proposed position and velocity update strategy in CSO are introduced as follows:

We assume that the swarm includes N particles. Then, the N particles are randomly divided into $N/3$ groups. In each group, three particles compete with each other based on the f_{SDE} value. Specifically, the particle with the smallest f_{SDE} value is selected as the winner particle, namely X_w, while other two particles are determined as X_{L1} and X_{L2}, respectively. Then, the velocities and positions for two loser particles (i.e., X_{L1} and X_{L2}) are updated as follows:

$$\begin{aligned}
V_{L_1}(t+1) &= r_1 V_{L_1}(t) + r_2(X_w(t) - X_{L_1}(t)) + \psi r_3(X_{L_2}(t) - X_{L_1}(t)) \\
V_{L_2}(t+1) &= r_1 V_{L_2}(t) + r_2(X_w(t) - X_{L_2}(t)) \\
X_{L_i}(t+1) &= X_{L_i}(t) + V_{L_i}(t+1) + r_1(V_{L_i}(t+1) - V_{L_i}(t))
\end{aligned} \tag{3}$$

where r_1, r_2 and r_3 are uniformly randomly distributed values in [0,1]. ψ is a parameter within [0,1] that controls the influence of X_{L2}. In this way, two-third of particles have been updated in each generation. That is, the convergence speed is accelerated by using the proposed position and velocity update strategy. After that, as shown in line 10, X_w, X_{L1} and X_{L2} are mutated by using polynomial-based mutation (PM) [25] and then all the mutated children are added into *Offspring* in line 11. At last, the environmental selection process [50] is performed on *Offspring* and P' to update the newly population P'. The above evolutionary process is terminated when the *fes* reaches the evaluating number t_1. Finally, the population P' is returned. Please note that the process of NormalOptimization_***CSO*** is the same as that of Weightedoptimization_***CSO***, excepting with different inputs as well as output. More details of the process of NormalOptimization_***CSO*** can be also found in [43].

3.3 A self-organizing mechanism in S-WOF

In order to adaptively adjust the evaluating numbers for weighted based optimization approach (t_1) and normal optimization method (t_2) more effectively, a self-organizing mechanism based on the current evolutionary state is proposed in S-WOF. Different from the original weighted based optimization framework (i.e., WOF) [43], where the value of t_1 and t_2 are fixed during the whole evolutionary process, t_1 and t_2 in the proposed self-organizing strategy can be adaptively adjusted based on the current evolutionary state. In this paper, the survival rate of the parents at the current *t*-th generation, denoted as SR_t, is applied to reflect the current evolutionary state. The details of SR_t are introduced as follows:

Let us assume that the population P includes *N* solutions (i.e., $p_1, p_2, p_3,\dots, p_N$) and $W = (w_1, w_2, \dots, w_k)$ includes *k* weight values. Then, W is performed on P to generate a new population P'. Afterward, an environmental selection process is performed on the union population $P \cup P'$ and then the newly updated population P' is obtained for the next generation. Hence, the survival rate SR_t is calculated as follows:

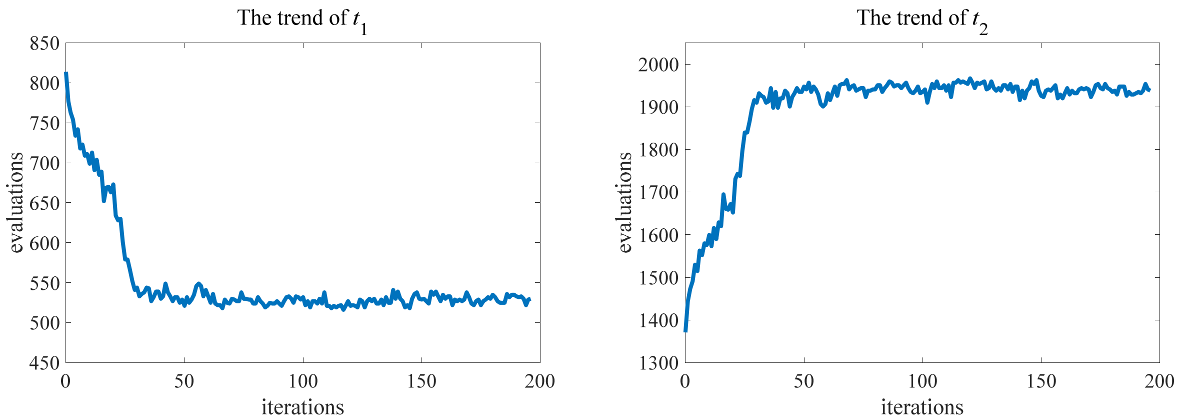
$$SR_t = \frac{count}{N} \tag{4}$$

where *count* represents the number of solutions from P that are survived in the newly population P'. *N* stands for the population size of the given problem and *t* indicates the current *t*-th generation. In this way, SR_t is used to reflect the current evolutionary state. Particularly, a small SR_t value indicates that the population is evolved dramatically. In other words, a small SR_t means the evolutionary state is in exploitation state, where many promising offspring are generated during this stage. On the other hand, a large SR_t value indicates that there are few promising children that are generated during this period. Hence, it is considered that the evolutionary state is in exploration state. Therefore, the current evolutionary state is applied to dynamically adjust the evaluating numbers, in which the evaluating numbers of weighted based optimization method (i.e., t_1) and normal optimization approach (i.e., t_2) are respectively calculated as follows:

$$t_1 = \alpha \times (2 - \frac{1}{e^{SR_t}}) \tag{5}$$

$$t_2 = \beta \times (1 + \frac{1}{e^{SR_t}}) \tag{6}$$

where α and β are respectively the initial values for t_1 and t_2. Please note that we set α=500 and β=1000 in this paper, which refers to the parameter settings in the original WOF framework [45]. Nevertheless, different from the WOF that keeps the evaluating numbers as a fixed value (i.e., t_1=500, t_2=1000), t_1 and t_2 are adaptively adjusted in the proposed self-organizing mechanism, which are guided by the current evolutionary state as well as the current survival rate at t-th generation. To have an intuitive observation on the changing tendency of t_1 and t_2, the graphs of t_1 and t_2 with the iterations are respectively plotted in Fig.1 (a) and Fig.1 (b), as follows:



(a) (b)

Fig. 1. The change tendency of (a) t_1 and (b) t_2 based on the survival rate that reflects the current evolution state: a high survival rate indicates the current evolution is in exploitation stage (a larger t_1 aims to accelerate convergence), whereas the opposite indicates the current evolution is in exploration stage (a larger t_2 aims for diversity maintenance).

As we can learn from Fig. 1 (a) and Fig. 1 (b), t_1 and t_2 change dramatically at the early evolutionary stage, as the evolutionary stage is in the exploitation state where there are many promising offspring that are generated to update the population. In other words, the survival rate SR_t is very small when the evolutionary stage is in exploitation state. Nevertheless, SR_t decreases slowly with the increasing of the iterations when the evolutionary stage switches the exploitation state to exploration state. Specifically, the evaluating numbers of weighted based optimization method t_1 is lager at early evolutionary stage when the population is in the exploitation state, in which more attentions are putted on the convergence speed. However, t_1 reaches a stable state at later evolutionary stage since the population gradually switches to the exploration state where more focuses are concentrated on diversity maintaining. Conversely, the evaluating numbers of normal optimization method t_2 shows an opposite trend to t_1, which is small at beginning and then rapidly increases when the population is in the exploitation state. Later, t_2 also reaches a stable state since the population is in a stagnation state, in which the diversity maintenance is emphasized. In this way, the evaluating numbers (i.e., t_1 and t_2) can be adaptively adjusted during the completely evolutionary process based on the current evolutionary state. That is, a dynamic trade-off is achieved in the proposed self-organizing mechanism. The pseudo-code of the proposed self-organizing mechanism is given in **Algorithm 3**.

Algorithm 3: t_1, t_2←***Self-organizing Mechanism*** (P', P)

Input: P', P
Output: t_1, t_2
count ← calculate the number of P in P';
SR_t ←update the survival rate by using Eq. (4);
t_1 ← update the evaluating numbers of weighted based optimization method by using Eq. (5);
t_2 ← update the evaluating numbers of normal optimization method by using Eq. (6);

4. Experimental results

In this section, several experiments are conducted to investigate the performance of the proposed S-WOF in solving LSMOPs. The superiority of the proposed S-WOF is verified by comparing it with four state-of-the-art MOEAs on LSMOPs, including WOF-NSGA-II [43], LMOCSO [36], DGEA [44] and LMEA [45]. A brief introduction of each compared algorithm is given as follows.

(1) WOF-NSGA-II [43]: WOF-NSGA-II proposes a weight-based optimization framework that applies a problem transformation technique to reduce the dimension of the search space.

(2) LMOCSO [36]: LMOCSO proposes a two-stage-based particle update strategy to update the position for each particle. At first, the velocity of each particle is updated by a leader particle and then the position of each particle is updated according to its updated velocity.

(3) DGEA [44]: DGEA proposes an adaptive offspring generation method. Firstly, it selects a balanced parent. Then, these parent solutions are used to construct direction vectors, which can reproduce promising offspring.

(4) LMEA [45]: LMEA proposes a decision variable clustering method, which divides the decision variables into convergence-related variables and diversity-related variables and then optimize separately.

4.1. Benchmark problems

In our experimental comparison, we have tested our S-WOF and other compared algorithms on nine benchmark problems. Regarding the adopted benchmark problems in our experiment, nine test problems (i.e., LSMOP1-LSMOP9 proposed in [51]) are used for experimental comparison. Among these problems, they have different characteristics. Specifically, LSMOP1-LSMOP4 are constructed by linear variable linkages in their ***PSs*** and a liner unit hyper-plane in their ***PFs***, while LSMOP5-LSMOP8 show nonlinear variable linkages in their ***PSs*** and a concave unit hyper-sphere in their ***PFs***. LSMOP9 has a nonlinear variable linkage in its ***PS*** and disconnected segments in its ***PF***.

4.2. The performance metrics

In our experimental comparison, we use IGD [52] and HV [10] to evaluate the performance of different algorithms, as they can simultaneously evaluate the performance of convergence and diversity.

(1) Inverted generational distance (IGD): We utilize IGD to measure the performance of solution set S' obtianed by each algorithm. Let us assume that *S* is a set of solutions uniformly distributed along the true ***PF*** in the objective space. Hence, the IGD value can be calculated as

$$IGD(S,S') = \frac{\sum_{i=1}^{|S|} d(S_i,S')}{|S|} \tag{7}$$

where $d(S_i, S^{'})$ is the minimum Euclidean distance from the point S_i to the individuals in population $S^{'}$, and $|S|$ represents the size of S. The true ***PF*** of the target problem is assumed available in advance when computing IGD. A lower value of $IGD(S, S^{'})$ indicates a better overall performance. Regarding the performance metric, the inverted generational distance (IGD) [52] is adopted to assess the performance of all algorithms in our experiment. Please note that 10000 reference points are sampled on each Pareto front of LSMOP1-LSMOP9 [51] for calculating the IGD value, which is based on the methods suggested in [53].

(2) Hypervolume (HV): Let us assume that S is the approximation to the true ***PF*** in the objective space, which is obtained by the algorithm. Let $z^r = (z_1^r, z_2^r, ..., z_m^r)^T$ be a reference point in the objective space, which can dominate any Pareto optimal objective vector. Then, HV can be defined as

$$HV(S) = VoI\left(\bigcup_{x \in S}\left[f_1(x), z_1^r\right] \times ... \times \left[f_m(x), z_m^r\right]\right) \tag{8}$$

where $VoI(.)$ denotes the Lebesgue measures. A higher value of HV indicates a better overall performance.

The mean and standard deviation of the IGD and HV values are recorded after 20 independent runs by each algorithm. The Wilcoxon rank sum test with a significance level of 0.05 is adopted to perform statistical analysis on the experimental results, where "+", "-" and "~" respectively indicate that the result obtained by another compared algorithm is significantly better than, worse than and similar to that obtained by the proposed S-WOF. Note that all experiments are performed on PlatEMO [54] and the source code of our designed S-WOF is available at https://github.com/xiaoyugege20/SWOF.git.

4.3. Experimental settings

(1) *Algorithms*: To have a fair comparison, the relative parameters of the compared algorithms are set according to the corresponding references [36, 43, 45, 46]. To be specific, in WOF-NSGA-II [43], the number of evaluations at each generation of original problem t_1, the number of evaluations of transferred problem t_2, and the number of groups γ are set to 1000, 500 and 4, respectively. The penalty parameter α of APD in LMOCSO is set to 2 [36]. The operation and RefNo are set to 3 and 10, respectively in DGEA [44]. The number of selected solutions *nSel* is set to 5, the number of perturbations for each selected solution in decision variable clustering *nPer* is set to 50 and the number of selected solutions in decision variable interaction analysis *nCor* is set to 5 in LMEA [45]. The penalty parameter α of APD, the number of evaluations at each generation of original problem t_1, the number of evaluations of transferred problem t_2, and the number of groups γ are set to 2, 1000, 500 and 4 in S-WOF, respectively.

(2) *Population Size*: The population size N is set to 300 in all algorithms.

(3) *Problems*: Regarding the number of objectives (M) for each test problem, we set M=2 and M=3 in our experiment. Regarding the number of decision variables (D), it is set from 100 to 1000 (i.e., n= (100, 200, 500, 1000)). In addition, the number of sub-components in each variable group n_k is set to 5.

(4) *Termination Criterion*: The maximum number of evaluations is set to $15000 \times D$, where D is the decision variable dimensions.

4.4. Comparisons between S-WOF and four competitive large-scale MOEAs

In this section, to verify the effectiveness of S-WOF, S-WOF is compared with four competitive large-scale MOEAs, including WOF-NSGA-II, LMOCSO, DGEA, and LMEA on solving nine benchmark problems (i.e., LSMOP1-LSMOP9). The scalability of S-WOF is verified for solving LSMOPs with the number of decision variables ranging from 100 to 1000. Finally, the final solutions obtained by S-WOF and four compared large-scale MOEAs can further verify the effectiveness of S-WOF. In addition, the CPU time comparison between S-WOF and four compared algorithms can verify that S-WOF shows a promising performance and an acceptable and reasonable CPU time simultaneously.

4.4.1 Comparison results on the two-objective LSMOPs

Table 1
The average IGD values and standard deviations obtained by S-WOF and its four competitors on two-objective LSMOP1–LSMOP9

Problem	M	D	WOF-NSGA-II	LMOCSO	DGEA	LMEA	S-WOF
LSMOP1	2	100	4.0340e-1(5.07e-2) -	3.0789e-3(2.10e-4) =	**2.4684e-3(1.56e-4)**+	1.4596e-2(1.05e-3)-	3.0201e-3(3.92e-4)
LSMOP1	2	200	4.2686e-1(5.34e-2) -	4.7952e-3(4.08e-4) -	**3.4613e-3(2.69e-4)**+	1.4194e-1(2.66e-1)-	3.9959e-3(3.96e-4)
LSMOP1	2	500	4.1022e-1(6.65e-2) -	1.4026e-2(1.13e-3) -	**6.4300e-3(8.37e-4)**=	5.2421e-2(9.33e-2)-	7.1473e-3(1.31e-3)
LSMOP1	2	1000	3.5829e-1(1.25e-1) -	9.8950e-2(2.03e-1) -	1.7658e-2(1.27e-3)-	1.4030e-1 (1.64e-1) -	**1.3992e-2(4.18e-3)**
LSMOP2	2	100	2.5868e-2(7.71e-3) -	4.8628e-2(1.66e-2) -	3.0564e-2(2.97e-2)-	1.2268e-1(4.41e-2)-	**1.0625e-2(9.71e-4)**
LSMOP2	2	200	1.7943e-2(5.64e-4) -	4.1959e-2(1.14e-2) -	1.4514e-2(1.63e-3) -	1.0960e-1(3.07e-2)-	**9.8940e-3(8.58e-4)**
LSMOP2	2	500	1.1086e-2(3.50e-4) -	3.1545e-2(9.80e-4) -	8.1514e-3(5.03e-4) -	6.0888e-2(5.07e-3)-	**5.2139e-3(1.20e-3)**
LSMOP2	2	1000	9.0983e-3 (6.40e-4) -	1.8046e-2(1.34e-3) -	5.2611e-3(2.27e-4) -	3.3683e-2 (3.16e-3) -	**3.0097e-3(5.99e-4)**
LSMOP3	2	100	**6.2328e-1(3.64e-2)** +	6.9037e-1(5.17e-2) =	1.6126e+0(1.29e+0)=	1.8030e+0(2.50e+0)-	7.0485e-1(7.08e-2)
LSMOP3	2	200	**6.5356e-1(2.63e-3)** +	7.8224e-1(2.31e-1) +	1.3457e+0(5.09e-2) -	3.0858e+0(4.66e+0)-	8.0283e-1(3.39e-2)
LSMOP3	2	500	**6.6001e-1(1.58e-2)** +	8.4000e-1(2.76e-1) +	1.3510e+0(1.93e-2) -	4.0250e+0(5.94e+0)-	8.6930e-1(3.92e-2)
LSMOP3	2	1000	**6.6949e-1(5.30e-3)** +	1.8542e+0(1.52e+0) =	1.3381e+0(3.58e-1) -	3.3320e+0 (2.75e+0) -	9.1834e-1(2.99e-2)
LSMOP4	2	100	5.0139e-2(2.49e-3) -	2.7645e-2(8.49e-4) -	1.4992e-2(3.20e-3) =	1.4160e+0(5.48e-2)-	**1.2014e-2(2.56e-3)**
LSMOP4	2	200	3.7614e-2(2.47e-3) -	1.6989e-2(4.12e-4) -	1.0228e-2 (1.21e-3) -	8.7735e-2(4.03e-2)-	**8.6726e-3(7.93e-4)**
LSMOP4	2	500	2.5569e-2(9.53e-4) -	1.1183e-2(5.40e-4) -	6.3799e-3 (8.63e-4) -	4.3447e-2(1.62e-2)-	**4.6142e-3(1.49e-4)**
LSMOP4	2	1000	2.4038e-2(7.57e-4) -	1.1229e-2(6.74e-4) -	4.8882e-3 (2.61e-4) -	2.1006e-2 (6.49e-4) -	**2.8382e-3(1.01e-4)**
LSMOP5	2	100	1.3462e-1(1.01e-1) -	**3.9376e-3(2.89e-4)** +	2.8370e-2 (1.13e-2) -	3.7641e-1(2.98e-1)-	2.8031e-2(6.91e-2)
LSMOP5	2	200	7.5729e-2(3.70e-2) =	**5.3896e-3(5.57e-4)** +	2.5781e-2(1.04e-2) =	4.1010e-1(2.73e-1)-	1.2193e-1(1.06e-1)
LSMOP5	2	500	3.9922e-2(1.64e-2) +	**9.4023e-3(2.72e-3)** +	3.0203e-2(1.65e-2) +	4.6232e-1(9.16e-2)-	2.1769e-1(8.51e-2)
LSMOP5	2	1000	3.2946e-2(6.03e-3) +	**1.9680e-2(4.18e-3)** +	5.3828e-2(5.15e-2) +	4.0997e-1 (1.58e-2) -	2.4601e-1(5.75e-2)
LSMOP6	2	100	**4.4301e-1(1.60e-1)** =	7.5751e-1(1.50e-2) -	6.6857e-1(5.86e-1) =	7.6759e-1(2.16e-1)-	4.9526e-1(1.00e-1)
LSMOP6	2	200	4.8044e-1(1.66e-1) =	7.6025e-1(6.66e-3) -	**4.2393e-1(1.30e-1)** +	7.7327e-1(1.07e-1)-	5.6040e-1(1.56e-1)
LSMOP6	2	500	**3.2969e-1(6.52e-2)** =	7.5450e-1(2.61e-3) -	3.4404e-1(2.05e-1) =	5.8865e-1(1.84e-1)-	3.9064e-1(1.99e-1)
LSMOP6	2	1000	3.0430e-1(5.24e-2) =	7.5340e-1(1.37e-3) -	**2.9228e-1(2.40e-1)** =	6.0161e-1 (2.37e-1) -	4.5854e-1(2.23e-1)
LSMOP7	2	100	**9.3666e-1(2.37e-1)** =	1.2986e+0(2.11e-1) -	1.5186e+0(6.60e-1) -	1.2704e+0(1.64e-1)-	9.6265e-1(1.40e-1)
LSMOP7	2	200	**9.3576e-1(1.87e-1)** +	1.6460e+0(3.25e-1) -	1.1820e+0(4.58e-1)=	1.3548e+0(2.25e-1)-	1.2978e+0(9.32e-2)
LSMOP7	2	500	**9.3822e-1(6.09e-2)** =	1.7418e+0(1.93e-1) -	1.0101e+0(4.19e-1)=	1.2194e+0(3.89e-1)=	1.0525e+0(2.33e-1)
LSMOP7	2	1000	1.0107e+0(1.61e-1) =	1.6297e+0(2.29e-1) -	2.0160e+0(5.40e-1) -	1.2447e+0 (4.00e-1) =	**9.0154e-1(2.59e-2)**
LSMOP8	2	100	1.2756e-1(2.89e-2) -	4.7421e-2(8.33e-3) -	5.6053e-2 (8.40e-3) -	9.0561e-2(2.51e-2)-	**4.7551e-3(2.60e-3)**
LSMOP8	2	200	6.1941e-2(1.83e-2) -	5.1568e-2(3.20e-3) -	4.7302e-2 (6.42e-3) -	6.5953e-2(2.93e-3)-	**4.6012e-3(6.80e-4)**
LSMOP8	2	500	4.2151e-2(1.42e-2) -	2.6046e-2(6.04e-4) -	3.7752e-2 (1.48e-2) -	4.1147e-2(1.97e-3)-	**8.3719e-3(1.55e-3)**
LSMOP8	2	1000	3.2790e-2(4.96e-3) -	1.3751e-2(8.75e-4) -	3.3468e-2 (9.40e-3) -	3.8178e-2 (7.09e-3) -	**7.0492e-3(2.71e-4)**
LSMOP9	2	100	8.1004e-1(1.17e-16) =	**1.5858e-2(2.85e-3)** +	4.4052e-1(3.17e-1) +	7.1036e-1(2.77e-1)=	8.1004e-1(1.17e-16)
LSMOP9	2	200	8.1004e-1(3.10e-16) -	9.3795e-2(6.21e-2) +	**7.9983e-2(3.83e-2)** +	6.8305e-1(2.97e-1)=	7.3049e-1(2.52e-1)
LSMOP9	2	500	8.0988e-1(1.91e-4) -	**1.0472e-1(9.80e-3)** +	2.2113e-1(2.05e-1) +	5.4352e-1(1.65e-1)-	4.5473e-1(1.91e-1)
LSMOP9	2	1000	8.0728e-1(1.11e-3) -	**5.9582e-2(3.57e-3)** =	9.2922e-2(4.40e-3) =	5.8055e-1 (2.02e-1) -	3.2233e-1(2.15e-1)
+/-/=			7/20/9	9/23/4	8/18/10	0/32/4	

Table 1 provides a comparison of results in terms of the IGD values obtained by each compared algorithm for two-objective LSMOP1-LSMOP9 with 100, 200, 500 and 1000 decision variables. Note that all the experimental results are obtained after 20 independent runs. As observed from Table 1, the proposed

S-WOF performed better than the other four competitors. Specifically, S-WOF can obtain the best results on 14 out of the 36 cases, while WOF-NSGA-II, LMOCSO, DGEA and LMEA only showed the best performance on 9, 7, 6, 0 out of 36 cases. From the one-by-one comparisons in the last row of Table 1, S-WOF performed better than WOF-NSGA-II, LMOCSO, DGEA and LMEA in 20, 23, 18 and 32 out of 36 cases, respectively, and it only underperformed in 7, 9, 8 and 0 out of 36 cases, respectively. According to the experimental results summarized in Table 1, S-WOF performed well in most adopted benchmark problems, including LSMOP1, LSMOP2, LSMOP4 and LSMOP8, while may show slight disadvantages in LSMOP3, LSMOP6 and LSMOP7 when compared to WOF-NSGA-II and show slight disadvantages in LSMOP1 when compared to DGEA. Regarding the performance of S-WOF in solving LSMOP3, LSMOP6 and LSMOP7 with multimodal landscapes, S-WOF performed poorly mainly due to the lack of diversity. Regarding the performance of S-WOF in solving LSMOP9 with discontinuous landscape, S-WOF behaved poorly because S-WOF lacks diversity. In other LSMOPs, S-WOF showed its superiority because of its strong convergence pressure by using the efficient CSO strategy and self-organizing mechanism. To conclude, the IGD results summarized in Table 1 have validated the effectiveness of the proposed S-WOF over other competitive MOEAs in most cases.

The statistical HV results of S-WOF and other compared algorithms are listed in Table 2. Some conclusions can be drawn from Table 2. The proposed S-WOF showed better overall performance than the other four competitors did. Specifically, S-WOF can obtain the best results on 13 out of the 36 cases, while WOF-NSGA-II, LMOCSO, DGEA and LMEA only showed the best on 5, 8, 6 and 2 out of 36 cases, respectively. From the one-by-one comparisons in the last row of Table 2, S-WOF performed better than WOF-NSGA-II, LMOCSO, DGEA and LMEA in 20, 17, 12 and 27 out of 36 cases, respectively, and it was only worse than WOF-NSGA-II, LMOCSO, DGEA and LMEA in 8, 10, 7, and 0 out of 36 cases, respectively. According to the HV results listed in Table 2, S-WOF performed well in most adopted benchmark problems, including LSMOP1, LSMOP2, LSMOP4 and LSMOP8, while may show slight disadvantages in LSMOP3 and LSMOP6 when compared to WOF-NSGA-II and show slight disadvantages in LSMOP1 and LSMOP6 when compared to DGEA. Regarding the performance of S-WOF in solving LSMOP3, LSMOP6 and LSMOP7 with multimodal landscapes, S-WOF performed poorly mainly due to the lack of diversity. Regarding the performance of S-WOF in solving LSMOP9 with discontinuous landscape, S-WOF performed poorly mainly because S-WOF lacks diversity. To conclude, the HV results summarized in Table 2 has validated the effectiveness of the proposed S-WOF over other competitive MOEAs.

Table 2
The average HV values and standard deviations obtained by S-WOF and its four competitors on two-objective LSMOP1–LSMOP9

Problem	M	D	WOF-NSGA-II	LMOCSO	DGEA	LMEA	S-WOF
LSMOP1	2	100	1.2612e-1 (3.48e-2) -	5.8223e-1 (2.69e-4) =	**5.8309e-1 (2.35e-4)** +	5.6476e-1(1.22e-3)-	5.8231e-1 (5.19e-4)
LSMOP1	2	200	1.0883e-1 (1.05e-2) -	5.8011e-1 (5.04e-4) -	**5.8169e-1 (3.63e-4)** +	4.7084e-1(1.93e-1)-	5.8111e-1 (4.93e-4)

LSMOP1	2	500	1.3684e-1 (3.71e-2) -	5.6906e-1 (1.36e-3) -	**5.7793e-1 (1.03e-3) =**	5.0666e-1(1.42e-1)-	5.7730e-1 (1.57e-3)
LSMOP1	2	1000	1.9319e-1 (1.16e-1) -	4.9247e-1 (1.61e-1) -	5.6427e-1 (1.52e-3) -	3.9619e-1 (2.16e-1) -	**5.6910e-1 (5.13e-3)**
LSMOP2	2	100	5.5361e-1 (9.67e-3) -	5.2518e-1 (2.20e-2) -	5.4896e-1 (3.56e-2) -	4.3091e-1(5.26e-2)-	**5.7305e-1 (1.15e-3)**
LSMOP2	2	200	5.6358e-1 (6.36e-4) -	5.3282e-1 (1.50e-2) -	5.6830e-1 (1.97e-3) -	4.4795e-1(3.69e-2)-	**5.7396e-1 (1.02e-3)**
LSMOP2	2	500	5.7202e-1 (4.16e-4) -	5.4567e-1 (1.18e-3) -	5.7593e-1 (6.09e-4) -	5.0795e-1(6.24e-3)-	**5.7958e-1 (1.45e-3)**
LSMOP2	2	1000	5.7451e-1 (7.54e-4) -	5.6345e-1 (1.70e-3) -	5.7942e-1 (2.74e-4) -	5.4278e-1 (3.94e-3) -	**5.8233e-1 (7.97e-4)**
LSMOP3	2	100	9.0486e-2 (4.74e-4) +	**9.0909e-2 (6.61e-7) +**	3.6347e-2 (4.69e-2) =	0.0000e+0(0.00e+0)-	3.7098e-2 (3.30e-2)
LSMOP3	2	200	**9.0533e-2 (4.70e-4) +**	7.9096e-2 (2.88e-2) +	0.0000e+0 (0.00e+0) =	0.0000e+0(0.00e+0)=	0.0000e+0 (0.00e+0)
LSMOP3	2	500	**9.0320e-2 (1.32e-3) +**	6.1524e-2 (4.29e-2) +	0.0000e+0 (0.00e+0) =	0.0000e+0(0.00e+0)=	0.0000e+0 (0.00e+0)
LSMOP3	2	1000	**8.7608e-2 (8.31e-3) +**	1.7647e-2 (3.72e-2) =	1.8183e-2 (3.83e-2) =	0.0000e+0(0.00e+0)=	0.0000e+0 (0.00e+0)
LSMOP4	2	100	5.0975e-1 (3.61e-3) -	5.5249e-1 (1.01e-3) -	5.6749e-1 (3.89e-3) =	4.0931e-1(5.54e-2)-	**5.7138e-1 (3.09e-3)**
LSMOP4	2	200	5.3290e-1 (3.04e-3) -	5.6533e-1 (5.03e-4) -	5.7332e-1 (1.49e-3) -	4.7429e-1(4.53e-2)-	**5.7540e-1 (9.53e-4)**
LSMOP4	2	500	5.5197e-1 (1.21e-3) -	5.7233e-1 (6.48e-4) -	5.7796e-1 (1.08e-3) -	5.2870e-1(1.92e-2)-	**5.8029e-1 (1.85e-4)**
LSMOP4	2	1000	5.5574e-1 (8.18e-4) -	5.7233e-1 (7.99e-4) -	5.7981e-1 (3.44e-4) -	5.5733e-1 (9.22e-4) -	**5.8253e-1 (1.40e-4)**
LSMOP5	2	100	2.5944e-1 (5.37e-2) -	**3.4646e-1 (4.57e-4) +**	3.3826e-1 (5.18e-3) =	1.7617e-1(1.11e-1)-	3.2661e-1 (5.41e-2)
LSMOP5	2	200	2.9056e-1 (2.14e-2) =	**3.4427e-1 (7.78e-4) +**	3.3532e-1 (4.74e-3) +	1.5410e-1(1.04e-1)-	2.4633e-1 (8.09e-2)
LSMOP5	2	500	3.0740e-1 (1.54e-2) +	**3.3874e-1 (3.30e-3) +**	3.2162e-1 (1.56e-2) +	8.7940e-2(8.61e-4)-	1.8959e-1 (5.35e-2)
LSMOP5	2	1000	3.0821e-1 (1.02e-2) +	**3.2328e-1 (6.25e-3) +**	3.0402e-1 (3.42e-2) +	8.5283e-2 (1.77e-3) -	1.7587e-1 (1.10e-2)
LSMOP6	2	100	**1.0920e-1 (2.41e-2) +**	5.2313e-2 (2.81e-2) =	4.7337e-2 (3.86e-2) =	6.9026e-3(2.18e-2)-	6.1366e-2 (2.47e-2)
LSMOP6	2	200	**1.1143e-1 (2.57e-2) +**	5.0841e-2 (1.26e-2) =	6.6206e-2 (3.41e-2) =	0.0000e+0(0.00e+0)-	6.6587e-2 (3.42e-2)
LSMOP6	2	500	1.1368e-1 (1.74e-2) =	6.2657e-2 (6.07e-3) -	**1.2319e-1 (7.68e-2) =**	4.9720e-4(1.57e-3)-	1.1337e-1 (4.69e-2)
LSMOP6	2	1000	1.3042e-1 (1.93e-2) =	6.6015e-2 (2.12e-3) -	**1.6386e-1 (8.04e-2) =**	3.1874e-2 (1.24e-2) -	1.1035e-1 (3.46e-2)
LSMOP7	2	100	0.0000e+0(0.00e+0) =	0.0000e+0(0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0(0.00e+0)=	0.0000e+0 (0.00e+0)
LSMOP7	2	200	0.0000e+0(0.00e+0) =	0.0000e+0(0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0(0.00e+0)=	0.0000e+0 (0.00e+0)
LSMOP7	2	500	0.0000e+0(0.00e+0) =	0.0000e+0(0.00e+0) =	2.9347e-3 (9.28e-3) =	**4.2655e-3(1.35e-2)=**	0.0000e+0 (0.00e+0)
LSMOP7	2	1000	0.0000e+0(0.00e+0) =	0.0000e+0(0.00e+0) =	0.0000e+0 (0.00e+0) =	**6.7436e-3 (1.57e-2) =**	0.0000e+0 (0.00e+0)
LSMOP8	2	100	2.5357e-1 (2.66e-2) -	2.9296e-1 (9.16e-3) -	2.7531e-1 (8.70e-3) -	2.4340e-1(3.11e-2)-	**3.4530e-1 (3.08e-3)**
LSMOP8	2	200	2.9568e-1 (1.39e-2) -	2.9145e-1 (5.43e-3) -	2.9321e-1 (1.22e-2) -	2.6329e-1(1.71e-3)-	**3.4503e-1 (1.02e-3)**
LSMOP8	2	500	3.0308e-1 (1.13e-2) -	3.2405e-1 (9.81e-4) -	3.2276e-1 (4.67e-3) -	2.9443e-1(2.31e-3)-	**3.3867e-1 (2.33e-3)**
LSMOP8	2	1000	3.0615e-1 (6.42e-3) -	3.3474e-1 (4.85e-4) -	3.2890e-1 (4.23e-3) -	2.9764e-1 (1.01e-2) -	**3.4046e-1 (3.80e-4)**
LSMOP9	2	100	9.0909e-2 (0.00e+0) =	**2.3501e-1 (1.76e-3) +**	1.6249e-1 (5.58e-2) +	1.0898e-1(6.52e-2)=	9.0909e-2 (0.00e+0)
LSMOP9	2	200	9.0909e-2 (6.51e-17) -	1.9776e-1 (2.78e-2) +	**2.0528e-1 (1.74e-2) +**	1.1425e-1(6.75e-2)=	1.0534e-1 (4.56e-2)
LSMOP9	2	500	9.0929e-2 (2.49e-5) -	**1.9214e-1 (4.15e-3) +**	1.6418e-1 (2.56e-2) =	1.4925e-1(3.88e-2)-	1.6751e-1 (3.51e-2)
LSMOP9	2	1000	9.1298e-2 (1.82e-4) -	**2.1319e-1 (1.62e-3) =**	1.9800e-1 (2.13e-3) =	1.4064e-1 (4.71e-2) -	1.9125e-1 (3.19e-2)
+/-/=			8/20/8	10/17/9	7/12/17	0/27/9	

To have an intuitive observation, some final solution sets obtained by each compared algorithm are plotted in Figs. 2-3 on two-objective LSMOP8 with 200 decision variables and two-objective LSMOP4 with 1000 decision variables, respectively. Note that the obtained solutions are marked as blue rhombus and the red lines indicate the true PFs that are plotted for performance comparison. Some conclusions can be easily drawn from these figures. Regarding the different test problems with various decision variables, all the final solution sets obtained by the proposed S-WOF are distributed evenly in these representative problems with different characteristics, while the final solution sets obtained by other competitors show worse distributions. Hence, it is reasonable to conclude that the proposed S-WOF shows an obvious superiority over other compared algorithms in solving LSMOP test problems with different decision variables.

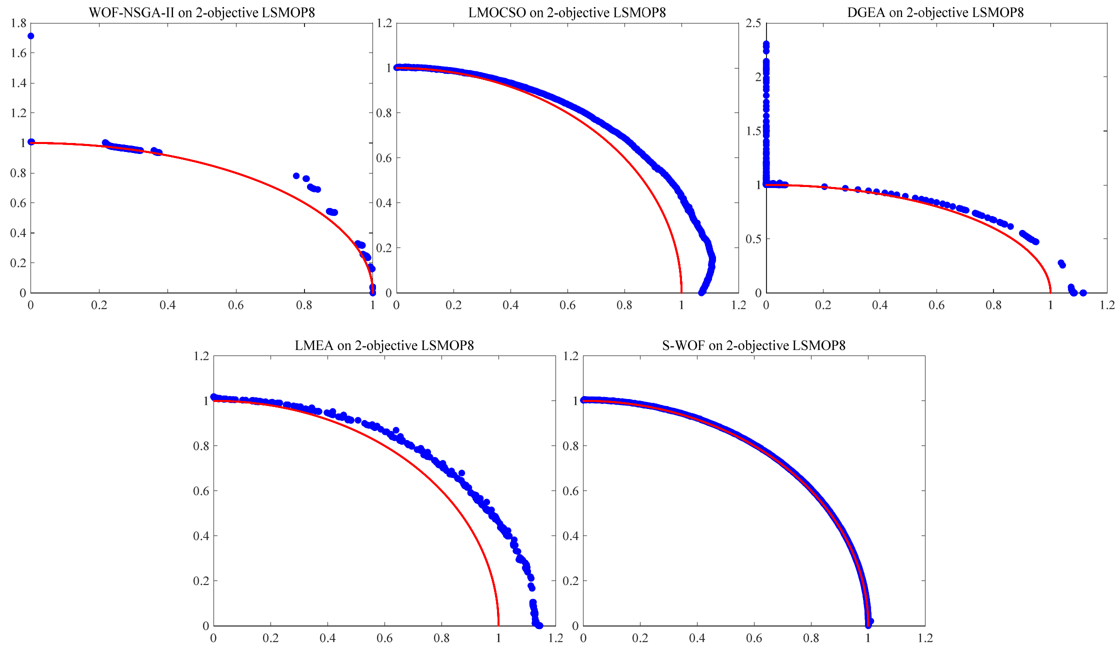


Fig. 2. The final non-dominated solutions obtained by all the compared algorithms on 2-objective LSMOP8 with 200 decision variables.

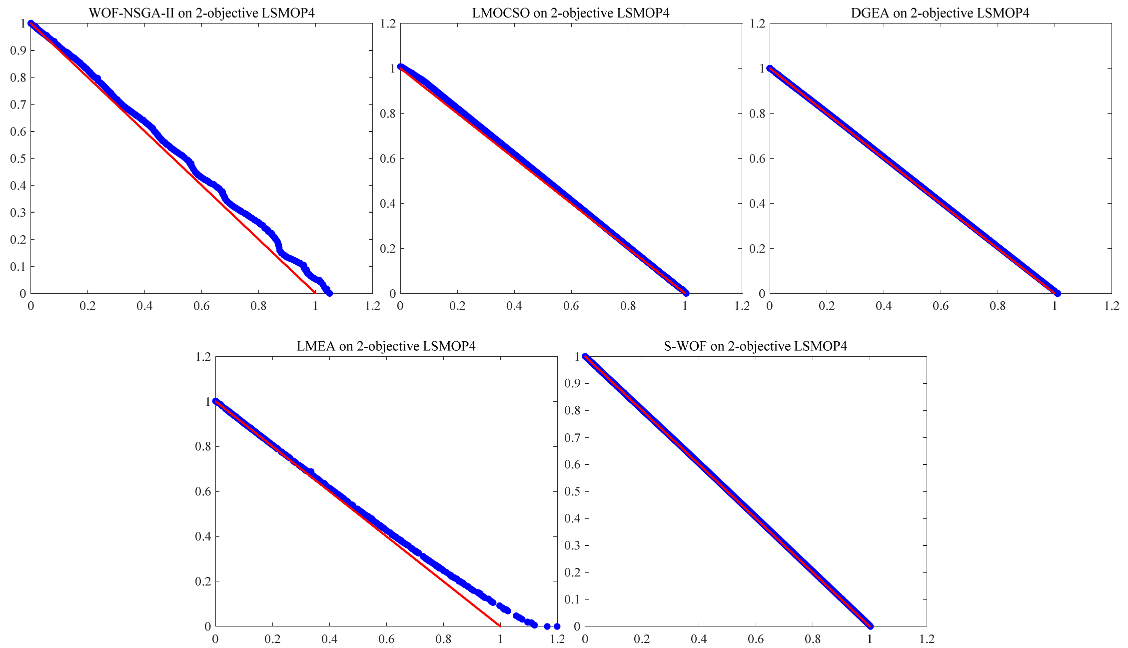


Fig. 3. The final non-dominated solutions obtained by all the compared algorithms on 2-objective LSMOP4 with 1000 decision variables.

4.4.2 Comparison results on the three-objective LSMOPs

Table 3 provides a comparison of results in terms of the IGD values obtained by each compared algorithm and S-WOF for three-objective LSMOP1-LSMOP9 with 100, 200, 500 and 1000 decision variables. It is obvious that the proposed S-WOF obtained better overall performance than the other four competitors did. Specifically, S-WOF can obtain the best results on 21 out of the 36 cases, while WOF-NSGA-II, LMOCSO, DGEA and LMEA only showed the best results on 2, 5, 4 and 4 out of 36 cases, respectively. From the one-by-one comparisons in the last row of Table 3, S-WOF performed better than WOF-NSGA-II, LMOCSO, DGEA and LMEA in 31, 29, 27 and 25 out of 36 cases, respectively, and it was only respectively worse in 2, 4, 5 and 4 out of 36 cases. According to the experiments in Table 3, S-WOF performed well in most adopted benchmark problems, including LSMOP1, LSMOP2, LSMOP4, LSMOP5 and LSMOP8, while may show slight disadvantages in LSMOP9 when compared to LMOCSO.

Regarding the performance of S-WOF in solving LSMOP9 with discontinuous landscape, S-WOF performed poorly mainly because S-WOF converged prematurely. Apart from the above cases, S-WOF showed its great superiority in most adopted cases, which mainly because S-WOF has excellent convergence pressure through using the proposed CSO strategy and self-organizing mechanism. Hence, the IGD results summarized in Table 3 have validated the effectiveness of the proposed S-WOF over other competitive large-scale MOEAs.

Table 3
The average IGD values and standard deviations obtained by S-WOF and its four competitors on three-objective LSMOP1–LSMOP9

Problem	M	D	WOF-NSGA-II	LMOCSO	DGEA	LMEA	S-WOF
LSMOP1	3	100	2.0648e-1(5.73e-3) -	6.2339e-2 (1.72e-2) -	1.8201e-1 (6.23e-2) -	4.4548e-2(1.72e-2) -	**3.1140e-2(2.06e-3)**
LSMOP1	3	200	2.0309e-1(3.40e-2) -	8.1250e-2 (9.16e-3) -	2.9000e-1 (2.14e-2) -	1.1382e-1(9.90e-2) -	**3.7918e-2(6.10e-3)**
LSMOP1	3	500	1.5672e-1(3.00e-2) -	9.7606e-2 (8.35e-3) -	3.2798e-1 (1.31e-2) -	8.8396e-2(7.14e-2) =	**4.9185e-2(9.28e-3)**
LSMOP1	3	1000	1.8669e-1(3.26e-2) -	1.0296e-1 (1.63e-2) -	3.4836e-1 (1.65e-2) -	1.0024e-1(9.96e-2) -	**4.2533e-2(1.09e-2)**
LSMOP2	3	100	1.3389e-1(1.09e-3) -	6.5988e-2 (2.15e-3) -	7.4268e-2 (1.44e-2) -	1.0095e-1(9.35e-2) =	**5.2411e-2(3.81e-3)**
LSMOP2	3	200	1.2580e-1(1.36e-3) -	5.4807e-2 (1.55e-3) -	4.7497e-2 (2.32e-3) -	7.0993e-2(2.91e-2) -	**3.9341e-2(5.40e-3)**
LSMOP2	3	500	6.7045e-2(1.04e-3) -	3.4033e-2 (2.98e-4) -	**3.2549e-2 (5.64e-4) =**	4.8686e-2(6.60e-3) -	3.2937e-2(1.56e-3)
LSMOP2	3	1000	4.8606e-2(1.81e-3) -	**2.6176e-2 (1.25e-3) =**	2.8493e-2 (5.95e-4) -	1.0426e-1(1.46e-1) -	2.7849e-2(4.35e-3)
LSMOP3	3	100	6.6250e-1(2.54e-1) =	7.5839e-1 (7.57e-2) -	1.0667e+0(1.01e+0) =	**5.3316e-1(3.86e-2) +**	6.0616e-1(2.25e-2)
LSMOP3	3	200	6.9291e-1(2.18e-1) =	8.1291e-1 (5.26e-2) -	7.5457e-1 (5.20e-1) =	**6.1258e-1(6.38e-2) =**	6.2223e-1(2.44e-2)
LSMOP3	3	500	**4.3681e-1(3.05e-2) +**	8.5004e-1 (2.84e-2) -	5.7525e-1 (2.04e-2) +	6.8341e-1(6.54e-2) -	6.3748e-1(2.13e-2)
LSMOP3	3	1000	**4.4579e-1(3.43e-2) +**	1.1305e+0 (8.91e-1) -	5.7737e-1 (1.47e-2) +	1.4682e+0(1.66e+0) -	6.4363e-1(9.43e-3)
LSMOP4	3	100	1.5040e-1(1.36e-2) -	1.1438e-1 (9.70e-3) -	1.5222e-1 (1.30e-2) -	1.1398e-1(5.03e-2) =	**8.9143e-2(2.92e-2)**
LSMOP4	3	200	2.0638e-1(9.07e-3) -	8.6293e-2 (9.98e-3) -	1.3521e-1 (7.00e-3) -	1.0293e-1(7.48e-2) =	**7.5910e-2(5.28e-3)**
LSMOP4	3	500	1.7219e-1(4.47e-3) -	6.5152e-2 (3.04e-3) -	8.1473e-2 (3.94e-3) -	**4.6602e-2(8.95e-4) +**	5.5658e-2(5.91e-3)
LSMOP4	3	1000	1.1947e-1(2.38e-3) -	4.7149e-2 (4.26e-3) -	5.0087e-2 (8.45e-4) -	6.9789e-2(8.53e-2) -	**4.2141e-2(1.73e-3)**
LSMOP5	3	100	3.6087e-1(2.18e-3) -	6.9232e-2 (1.36e-2) -	3.3621e-1 (7.62e-2) -	1.6732e+0(1.09e+0) -	**4.5606e-2(1.03e-2)**
LSMOP5	3	200	3.1994e-1(6.59e-2) -	2.0585e-1 (9.13e-2) -	3.8010e-1 (1.97e-2) -	2.4991e+0(2.52e+0) -	**5.5774e-2(1.50e-2)**
LSMOP5	3	500	3.5782e-1(2.22e-2) -	3.7384e-1 (7.57e-2) -	3.8793e-1 (3.55e-2) -	6.3834e+0(3.20e+0) -	**7.7927e-2(3.55e-2)**
LSMOP5	3	1000	3.7203e-1(4.70e-2) -	6.4303e-1 (1.70e-1) -	4.0600e-1 (4.51e-2) -	8.0646e+0(4.13e+0) -	**9.1555e-2(2.83e-2)**
LSMOP6	3	100	8.7401e-1(4.47e-2) -	8.3758e-1 (3.94e-2) -	7.2501e-1 (1.34e-1) =	1.0349e+0(1.51e-1) -	**6.7694e-1(4.21e-2)**
LSMOP6	3	200	1.1148e+0(3.10e-3) -	9.9895e-1 (1.97e-1) -	**7.4572e-1 (1.93e-1) +**	8.0431e+2(9.36e+2) -	9.3359e-1(5.67e-2)
LSMOP6	3	500	1.2416e+0(1.29e-3) -	1.0013e+0 (4.11e-1) =	**7.6006e-1 (2.95e-1) +**	7.9835e+2(2.52e+3) -	9.5501e-1(2.29e-1)
LSMOP6	3	1000	1.2832e+0(5.91e-4) -	1.1732e+0 (4.45e-1) =	**7.4288e-1 (2.53e-1) +**	1.5359e+0(3.19e-1) -	1.1527e+0(1.81e-1)
LSMOP7	3	100	7.7249e-1(2.16e-2) -	9.4547e-1 (1.44e-3) -	9.8289e-1 (5.08e-2) -	2.2685e+0(9.45e-1) -	**6.4510e-1(3.10e-2)**
LSMOP7	3	200	8.4353e-1(1.90e-2) -	9.4593e-1 (1.01e-6) -	1.1481e+0 (1.11e-1) -	1.6182e+0(3.06e-1) -	**7.0139e-1(4.81e-2)**
LSMOP7	3	500	8.4735e-1(7.81e-3) -	9.4593e-1 (6.70e-7) -	1.1206e+0 (2.17e-1) -	8.7350e-1(1.28e-1) -	**7.6147e-1(4.71e-2)**
LSMOP7	3	1000	8.4227e-1(3.25e-4) -	9.4593e-1 (1.37e-5) -	1.1199e+0 (3.06e-1) -	**6.0539e-1(7.22e-2) +**	7.4680e-1(4.38e-2)
LSMOP8	3	100	3.1954e-1(5.58e-2) -	1.0564e-1 (7.33e-3) -	1.8770e-1 (7.28e-2) -	1.5494e-1(8.08e-2) -	**3.9304e-2(6.70e-3)**
LSMOP8	3	200	1.6598e-1(3.00e-2) -	9.8161e-2 (6.46e-3) -	8.8078e-2 (7.56e-3) -	1.3212e-1(1.21e-2) -	**3.4280e-2(2.70e-3)**
LSMOP8	3	500	1.5045e-1(7.83e-3) -	6.8204e-2 (1.62e-3) -	7.3967e-2 (3.16e-3) -	7.4491e-2(6.47e-3) -	**3.1876e-2(1.16e-3)**
LSMOP8	3	1000	1.4587e-1(1.29e-2) -	5.6712e-2 (4.37e-3) -	1.6727e-1 (2.77e-1) -	5.4653e-2(1.15e-3) -	**3.2483e-2(2.62e-3)**
LSMOP9	3	100	1.1450e+0(8.52e-6) =	**1.2970e-1 (6.77e-2) +**	1.2999e+0 (3.69e-1) -	6.2784e-1(2.81e-1) +	8.6919e-1(2.91e-1)
LSMOP9	3	200	1.1450e+0(3.36e-6) -	**1.6636e-1 (1.10e-1) +**	1.4096e+0 (4.71e-1) -	6.3617e-1(2.82e-1) =	7.0288e-1(2.33e-1)
LSMOP9	3	500	1.1450e+0(4.48e-6) -	**2.2201e-1 (5.91e-2) +**	1.2503e+0 (5.96e-1) -	6.4108e-1(4.34e-1) -	5.9239e-1(5.65e-5)
LSMOP9	3	1000	1.1449e+0(4.92e-6) -	**2.6135e-1 (1.74e-1) +**	1.0411e+0 (2.60e-1) -	8.4397e-1(5.72e-1) =	5.9243e-1(2.27e-4)
+/-/=			2/31/3	4/29/3	5/27/4	4/25/7	

The statistical HV results of each compared algorithm and S-WOF are listed in Table 4. Some conclusions can be drawn from this table. The proposed S-WOF obtained better overall performance than the other four competitors did. Specifically, S-WOF could obtain the best results on 17 out of the 36 cases, while WOF-NSGA-II, LMOCSO, DGEA and LMEA only showed the best results on 4, 9, 5 and 1 out of 36 cases, respectively. From the one-by-one comparisons in the last row of Table 4, S-WOF performed better than WOF-NSGA-II, LMOCSO, DGEA and LMEA in 28, 23, 26 and 23 out of 36 cases, respectively, and it was only worse than WOF-NSGA-II, LMOCSO, DGEA and LMEA in 3, 5, 7 and 2 out of 36 cases, respectively. Based on the results in Table 4, S-WOF performed well in most adopted benchmark problems, including LSMOP1, LSMOP2, LSMOP4, LSMOP5 and LSMOP8, while may show slight disadvantages in LSMOP3 when compared to WOF-NSGA-II, in LSMOP6 when compared to DGEA, and in LSMOP7 and LSMOP9 when compared to LMOCSO. Regarding the performance of S-WOF in solving LSMOP3, LSMOP6, LSMOP7 with multimodal landscape, S-WOF performed poorly mainly due to the lack of diversity. Regarding the performance of S-WOF in solving LSMOP9 with discontinuous landscape, S-WOF performed poorly because S-WOF converged prematurely. Therefore, the HV results summarized in Table 4 have validated the effectiveness of the proposed S-WOF over other competitors.

Table 4
The average HV values and standard deviations obtained by S-WOF and its four competitors on three-objective LSMOP1–LSMOP9

Problem	M	D	WOF-NSGA-II	LMOCSO	DGEA	LMEA	S-WOF
LSMOP1	3	100	6.1851e-1 (6.46e-3) -	8.0314e-1 (2.57e-2) -	6.6942e-1 (9.54e-2) -	8.2121e-1 (1.36e-2) -	**8.4329e-1 (2.09e-3)**
LSMOP1	3	200	6.2651e-1 (4.67e-2) -	7.7255e-1 (1.75e-2) -	4.9676e-1 (3.05e-2) -	7.2773e-1 (1.20e-1) -	**8.3671e-1 (5.82e-3)**
LSMOP1	3	500	6.8836e-1 (5.00e-2) -	7.4498e-1 (1.23e-2) -	4.4564e-1 (1.16e-2) -	7.3676e-1 (1.22e-1) -	**8.2499e-1 (9.52e-3)**
LSMOP1	3	1000	6.7508e-1 (2.61e-2) -	7.3940e-1 (2.65e-2) -	4.1852e-1 (1.27e-2) -	7.4204e-1 (1.41e-1) -	**8.3135e-1 (1.17e-2)**
LSMOP2	3	100	7.2926e-1 (1.91e-3) -	8.0733e-1 (2.11e-3) -	7.9890e-1 (1.80e-2) -	7.6819e-1 (8.66e-2) =	**8.2349e-1 (3.45e-3)**
LSMOP2	3	200	7.3618e-1 (1.58e-3) -	8.1807e-1 (1.43e-3) -	8.2619e-1 (2.61e-3) -	7.9716e-1 (2.91e-2) -	**8.3495e-1 (5.31e-3)**
LSMOP2	3	500	8.0165e-1 (1.30e-3) -	8.3897e-1 (3.45e-4) -	**8.4220e-1 (5.54e-4) +**	8.2287e-1 (6.52e-3) -	8.4099e-1 (1.61e-3)
LSMOP2	3	1000	8.2176e-1 (1.24e-3) -	**8.4804e-1 (1.24e-3) =**	8.4772e-1 (8.57e-4) =	7.8828e-1 (1.05e-1) -	8.4626e-1 (6.24e-3)
LSMOP3	3	100	**2.5360e-1 (2.18e-1) =**	9.3259e-2 (3.30e-3) -	1.3849e-1 (9.25e-2) =	2.4575e-1 (8.22e-2) +	1.0857e-1 (2.12e-2)
LSMOP3	3	200	**2.0995e-1 (1.60e-1) =**	9.1249e-2 (8.61e-4) -	1.4973e-1 (6.52e-2) +	1.7005e-1 (6.09e-2) +	9.7393e-2 (4.79e-3)
LSMOP3	3	500	**3.9571e-1 (4.53e-2) +**	9.1198e-2 (7.69e-4) -	1.7378e-1 (3.96e-3) +	1.2023e-1 (5.67e-2) =	9.5455e-2 (4.26e-3)
LSMOP3	3	1000	**3.8311e-1 (5.05e-2) +**	8.1818e-2 (2.87e-2) -	1.6909e-1 (6.30e-3) +	4.0821e-3 (6.16e-3) -	9.5725e-2 (2.99e-3)
LSMOP4	3	100	6.8695e-1 (1.71e-2) -	7.6218e-1 (9.95e-3) -	7.1764e-1 (1.62e-2) -	7.3279e-1 (7.62e-2) -	**7.8749e-1 (2.78e-2)**
LSMOP4	3	200	6.2843e-1 (8.27e-3) -	7.9092e-1 (9.04e-3) -	7.3907e-1 (7.98e-3) -	7.5349e-1 (8.77e-2) -	**8.0270e-1 (4.43e-3)**
LSMOP4	3	500	6.7760e-1 (5.19e-3) -	8.1050e-1 (2.92e-3) -	7.9450e-1 (4.27e-3) -	**8.2084e-1 (1.33e-3) =**	8.2011e-1 (6.11e-3)
LSMOP4	3	1000	7.4529e-1 (2.98e-3) -	8.2705e-1 (3.92e-3) -	8.2446e-1 (9.25e-4) -	8.0117e-1 (8.14e-2) =	**8.3182e-1 (1.89e-3)**
LSMOP5	3	100	3.7787e-1 (1.33e-2) -	5.2383e-1 (1.56e-2) -	3.7385e-1 (5.54e-2) -	1.3522e-1 (2.27e-1) -	**5.4801e-1 (1.62e-2)**
LSMOP5	3	200	3.8222e-1 (4.40e-2) -	3.7786e-1 (9.41e-2) -	3.5870e-1 (3.33e-2) -	2.2403e-1 (2.46e-1) -	**5.3637e-1 (1.85e-2)**
LSMOP5	3	500	3.6088e-1 (1.69e-2) -	2.1569e-1 (4.42e-2) -	3.5342e-1 (4.92e-2) -	4.9866e-2 (1.06e-1) -	**5.1072e-1 (3.99e-2)**
LSMOP5	3	1000	3.6618e-1 (1.84e-2) -	1.0477e-1 (1.54e-2) -	3.3449e-1 (6.05e-2) -	6.2012e-2 (1.35e-1) -	**4.9347e-1 (3.23e-2)**
LSMOP6	3	100	2.2492e-3 (1.99e-3) +	2.2264e-3 (4.80e-3) =	**2.8010e-2 (2.08e-2) +**	2.5108e-3 (7.94e-3) =	0.0000e+0 (0.00e+0)
LSMOP6	3	200	0.0000e+0(0.00e+0) =	1.3536e-2 (4.28e-2) =	**4.2750e-2 (2.03e-2) +**	0.0000e+0(0.00e+0) =	5.6838e-3 (1.80e-2)
LSMOP6	3	500	0.0000e+0 (0.00e+0) -	3.2743e-2 (5.47e-2) =	**6.0422e-2 (2.14e-2) =**	0.0000e+0 (0.00e+0) -	3.6736e-2 (3.90e-2)
LSMOP6	3	1000	0.0000e+0(0.00e+0) =	1.3636e-2 (2.90e-2) =	**6.9174e-2 (2.43e-2) +**	5.9012e-3 (1.87e-2) =	1.4556e-2 (3.07e-2)
LSMOP7	3	100	7.5820e-3 (1.65e-2) -	**9.0906e-2 (1.20e-6) +**	9.0885e-3 (2.87e-2) -	0.0000e+0 (0.00e+0) -	8.5523e-2 (1.50e-3)
LSMOP7	3	200	0.0000e+0 (0.00e+0) -	**9.0907e-2 (1.03e-6) +**	0.0000e+0 (0.00e+0) -	2.3570e-3 (7.45e-3) -	8.7303e-2 (1.72e-3)
LSMOP7	3	500	1.8684e-3 (5.33e-3) -	**9.0905e-2 (1.13e-6) +**	0.0000e+0 (0.00e+0) -	2.3724e-5 (7.50e-5) -	8.8637e-2 (9.31e-4)
LSMOP7	3	1000	1.6212e-2 (1.11e-3) -	**9.0895e-2 (3.09e-5) +**	8.8960e-3 (2.23e-2) -	1.0874e-2 (3.76e-3) -	8.9415e-2 (3.36e-4)
LSMOP8	3	100	3.9695e-1 (1.98e-2) -	4.6438e-1 (1.26e-2) -	4.0581e-1 (4.68e-2) -	3.7311e-1 (1.25e-1) -	**5.6026e-1 (8.23e-3)**
LSMOP8	3	200	4.2093e-1 (2.07e-2) -	4.7288e-1 (6.24e-3) -	4.9972e-1 (1.19e-2) -	3.9860e-1 (1.77e-2) -	**5.6724e-1 (3.89e-3)**
LSMOP8	3	500	4.2909e-1 (8.66e-3) -	5.1086e-1 (2.93e-3) -	5.2079e-1 (3.47e-3) -	4.8982e-1 (9.39e-3) -	**5.7092e-1 (2.10e-3)**
LSMOP8	3	1000	4.3785e-1 (1.53e-2) -	5.2500e-1 (6.82e-3) -	4.7409e-1 (1.41e-1) -	5.2452e-1 (1.65e-3) -	**5.7052e-1 (3.25e-3)**
LSMOP9	3	100	1.4772e-1 (1.37e-6) =	**2.3117e-1 (4.15e-2) +**	4.9345e-2 (5.15e-2) -	1.8242e-1 (2.15e-2) =	1.6786e-1 (2.16e-2)
LSMOP9	3	200	1.4772e-1 (1.13e-6) -	**2.1173e-1 (6.18e-2) =**	2.5876e-2 (1.50e-2) -	1.7553e-1 (2.00e-2) =	1.8048e-1 (1.75e-2)
LSMOP9	3	500	1.4771e-1 (1.96e-6) -	**1.7946e-1 (3.19e-2) =**	6.0559e-2 (7.13e-2) -	1.6217e-1 (5.79e-2) =	1.8870e-1 (4.04e-5)
LSMOP9	3	1000	1.4771e-1 (1.28e-6) -	**1.6902e-1 (8.12e-2) =**	6.3297e-2 (5.47e-2) -	1.4713e-1 (7.08e-2) =	1.8868e-1 (1.35e-4)

+/-/=			3/28/5	5/23/8	7/26/3	2/23/11	

For further observation, some final solution sets with the best IGD values from all 20 runs are plotted in Figs. 4-5 for the nondominated solutions obtained by the four compared MOEAs and S-WOF on three-objective LSMOP2 with 100 decision variables and three-objective LSMOP8 with 500 decision variables. Note that the blue and red dots in figures are the optimal solutions obtained by each compared algorithm and the true PFs, respectively. Some conclusions can be easily drawn from these figures. Regarding the different test problems with various decision variables, all the final solution sets obtained by the proposed S-WOF are distributed evenly in these representative problems with different characteristics, while the final solution sets obtained by other competitors show poor distributions. Hence, it is reasonable to conclude that the proposed S-WOF shows an obvious superiority over other compared algorithms in solving LSMOP test problems with different decision variables.

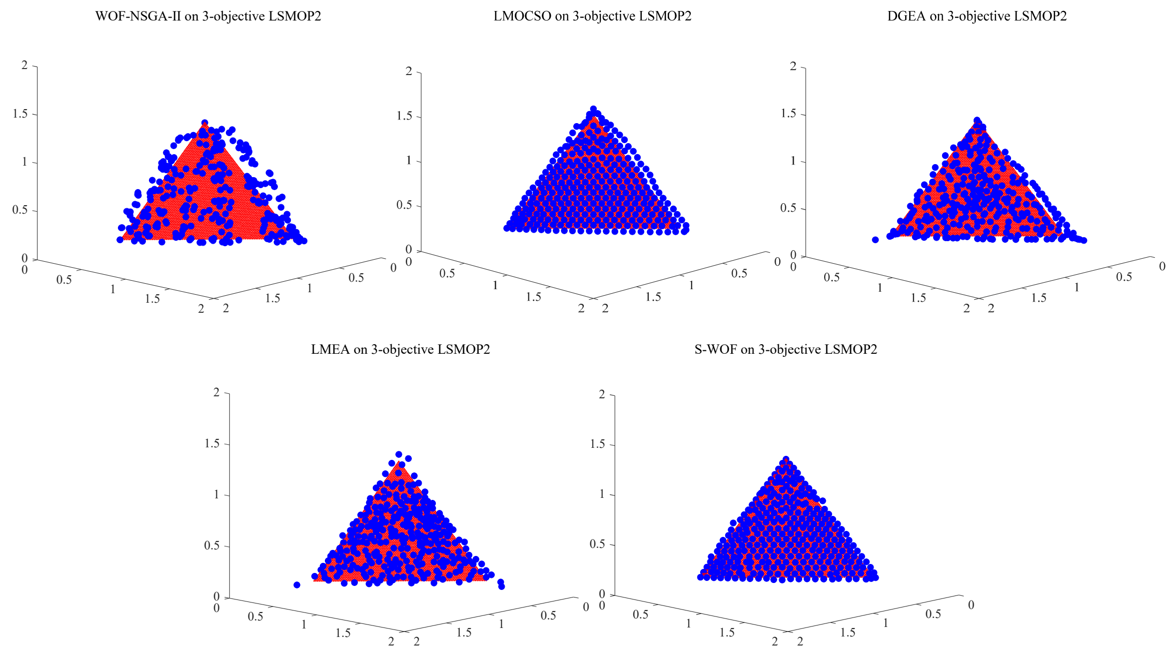


Fig. 4. The final non-dominated solutions obtained by all compared algorithms on 3-objective LSMOP2 with 100 decision variables.

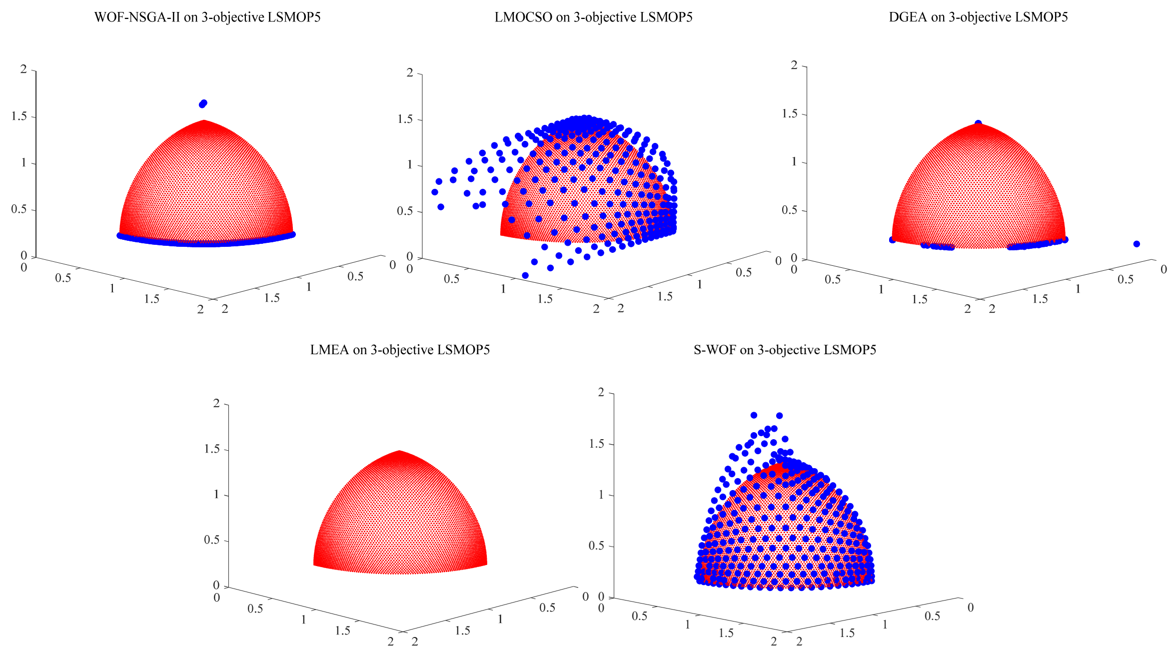


Fig. 5. The final non-dominated solutions obtained by all compared algorithms on 3-objective LSMOP5 with 500 decision variables

4.4.3 Comparison on CPU running time.

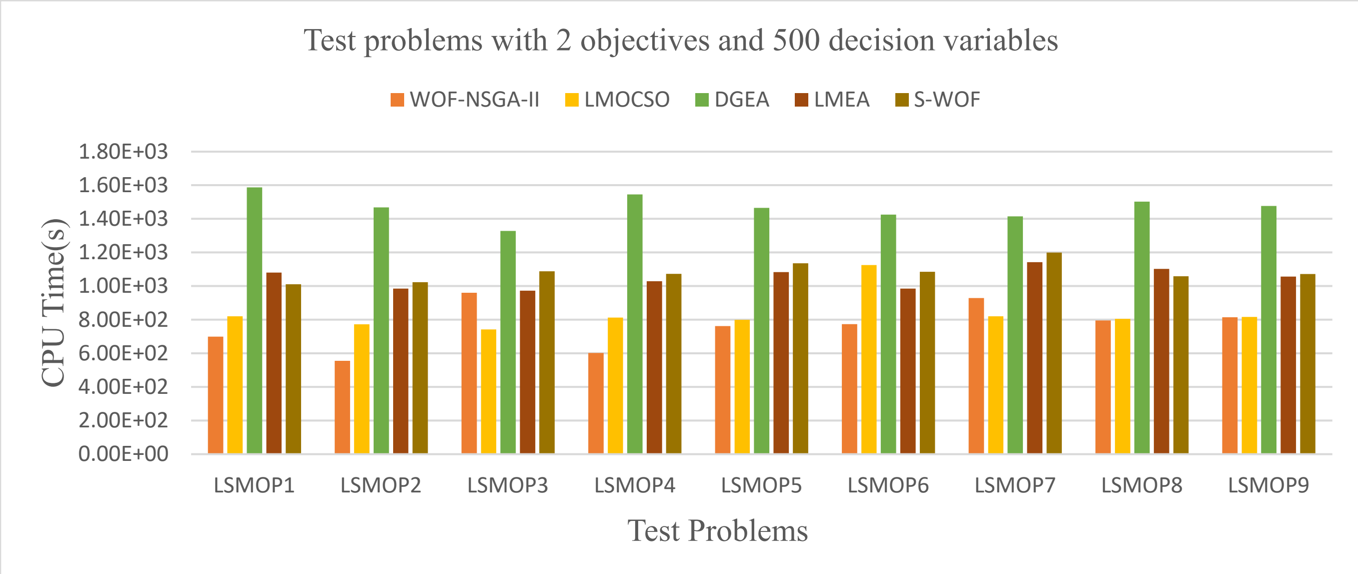


Fig. 6. Average CPU running time of all compared algorithms on all test problems with 2 objectives and 500 decision variables.

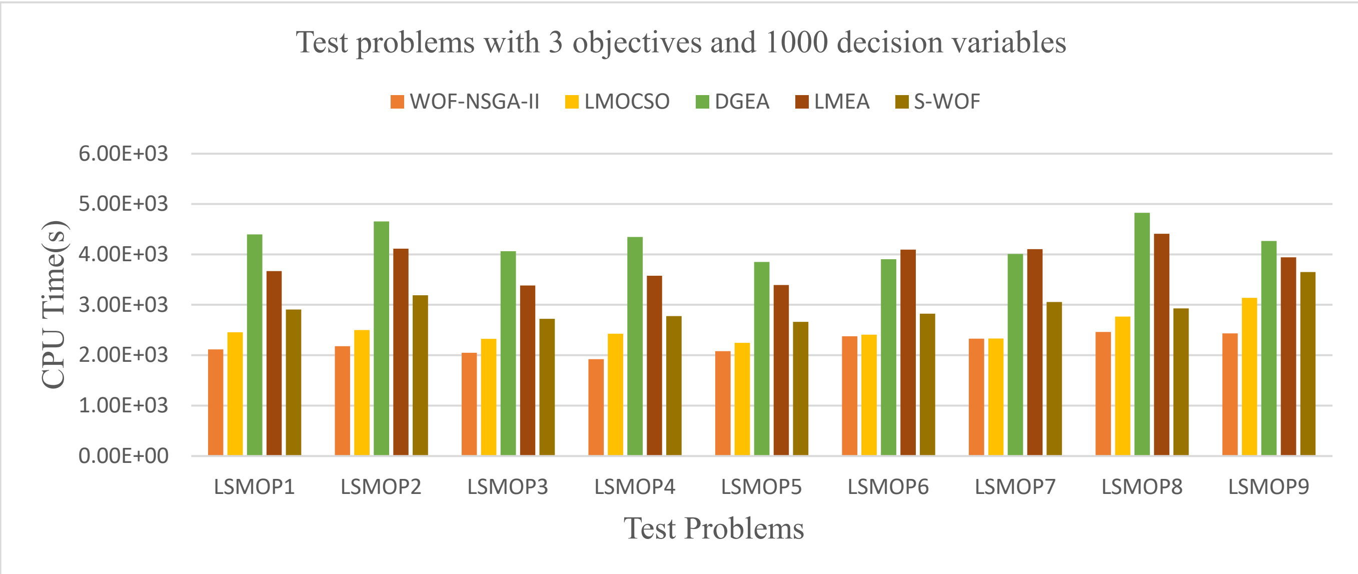


Fig. 7. Average CPU running time of all compared algorithms on all test problems with 3 objectives and 1000 decision variables.

The CPU running time is also an important criterion to evaluate the efficiency of an algorithm. Therefore, we further investigate the computational efficiency of our proposed method and other compared large-scale MOEAs. Figs. 6-7 provide the average CPU running time of each compared algorithm after 20 independent runs on all LSMOP test problems with two objectives and 500 decision variables and all LSMOP test problems with three objectives and 1000 decision variables (in seconds: s), respectively. We can draw following conclusions from these figures:

First, WOF-NSGA-II and LMOCSO showed some advantages over our proposed method in most adopted test problems. The main reason behind this is that these two algorithms were implemented in a simple MOEA framework, where WOF-NSGA-II adopted a simple non-dominated sorting genetic algorithm and LMOCSO adopted a basic CSO variant to optimize the LMOPs with large-scale variables. Despite these two methods showed some advantages in terms of the CPU running time, as learned from the average IGD results in Table 1 and Table 2, such a simple MOEA framework adopted in WOF-NSGA-II and LMOCSO may not be sufficient to address the various LMOPs in the huge search space due to its weak search ability to some extent.

Second, when comparing to LMEA, the CPU time of our proposed method was similar to that of LMEA for 2-objective LSMOP problems with 500 decision variables, while it showed advantages over LMEA when solving 3-objective LSMOP problems with 1000 decision variables. The reason behind this is that LMEA, as a method based on the decision variable analysis, requires more computation time to conduct decision variable analysis as the dimensionality of decision variables increases.

Third, DGEA performed worst in most adopted cases, because DGEA need to select the diversity-related solutions and convergence-related solutions as the balanced parent population before each iteration of evolution, which consumes a significant amount of time.

Last, when considering the CPU time of our proposed method, S-WOF showed slight disadvantages compared to WOF-NSGA-II and LMOCSO in most adopted cases. Here, we further investigate the main reasons behind this. Compared to WOF-NSGA-II, the main difference between S-WOF and WOF-NSGA-II is that the evaluating numbers for the weighted based optimization and original optimization in WOF-NSGA-II are fixed, while the evaluating numbers in our proposed S-WOF are dynamically adjusted according to the current evolutionary state, aiming to achieve a good tradeoff between diversity and convergence. Therefore, compared to WOF-NSGA-II, although our proposed S-WOF needs to consume a small amount of extra time for monitoring the current evolutionary state and adjustment of the evaluating numbers for the weighted based optimization and original optimization, it can achieve a better balance between convergence and diversity. On the other hand, compared to LMOCSO, one of the main difference between LMOCSO and S-WOF is the adopted optimizer during the evolutionary search. Specifically, LMOCSO adopted a basic CSO variant as the optimizer to perform the evolutionary search, where only half of particles are evolved in each iteration. However, such a basic CSO variant may show slow convergence when solving LMOPs with large-scale decision variables. To alleviate this issue, this paper proposed a new efficient CSO variant that allows two-thirds of particles to be evolved in each iteration, aiming to speed up the convergence. Therefore, compared to LMOCSO, although our proposed S-WOF consumes a small amount of extra time to evolve more particles in each iteration, it has a faster convergence speed. Since there is no free lunch, we sometimes need to sacrifice a small amount of time while improving the performance of the algorithm. Fortunately, compared to WOF-NSGA-II and LMOCSO, it can be seen from the experimental results that the performance of our method in solving various LMOPs has been obviously improved, while the CPU times of our method and these two competitors are still at the same and acceptable level.

Summarily, compared with these current competitive large-scale MOEAs, extensive experimental results show that our proposed S-WOF can achieve a good tradeoff between performance and execution efficiency. That is, S-WOF not only obtains promising performance in solving most adopted LMOPs, but also exhibits an acceptable and reasonable execution efficiency.

The main contribution and concern of this paper is to improve the performance of algorithm in solving LMOPs by designing some efficient techniques, including a self-organizing mechanism for balancing the diversity and convergence and a new CSO variant for further accelerating convergence. Of course, how to better guarantee the performance of the algorithm while further improving the efficiency of the algorithm execution is also a point we need to consider in the future. Moreover, we note that different programming styles may also have a significant impact on the CPU running time of the algorithms to some extent. Thus, we will attempt to adopt a more efficient coding style for programming when designing algorithms in future work.

5. Conclusions and future work

In this paper, we proposed a self-organizing weighted optimization based framework, termed S-WOF, which applies a self-organizing mechanism to adaptively adjust the evaluating numbers of weighted based optimization method and normal optimization method. Specifically, regarding the number of evaluations for the weighted based optimization approach (t_1), it becomes larger when the population is in the exploitation state, which aims to accelerate the convergence speed, while t_1 is gradually diminished when the population switches to the exploration state, in which more attentions are put on the diversity. On the other hand, regarding the number of evaluations for the normal optimization method (t_2), it shows an opposite trend to t_1, which becomes smaller during the exploitation state and gradually increases later. In this way, a dynamic trade-off between convergence and diversity is achieved in the proposed S-WOF. Moreover, an efficient CSO variant is designed in S-WOF to further improve the search ability in the large-scale search space. As illustrated in the experimental comparison and analysis, our proposed S-WOF has shown superiority on solving most benchmark problems adopted in this paper when compared to four state-of-the-art MOEAs (i.e., WOF-NSGA-II, LMOCSO, DGEA, and LMEA).

Although the experimental results have validated the effectiveness of our proposed method over several competitive large-scales MOEAs, with the rapid development of the field of large-scale optimization, there is still tremendous room for further research and improvement. First, designing a more computationally efficient framework as well as improving the robustness of our method in solving various LMOPs are on the agenda of our future work. Second, extending our method to solve large-scale many-objective optimization problems (LMaOPs) and some practical applications are also on the agenda of our future work.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 61876110, the Joint Funds of the NSFC under Key Program Grant U1713212, 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] C. Ramirez Atencia, J. Del Ser, D. Camacho, Weighted strategies to guide a multi-objective evolutionary algorithm for multi-UAV mission planning, Swarm and Evolutionary Computation, 44 (2019) 480-495.

[2] M.W. Przewozniczek, P. Dziurzanski, S. Zhao, L.S. Indrusiak, Multi-objective parameter-less population pyramid for solving industrial process planning problems, Swarm and Evolutionary Computation, 60 (2021) 100773.

[3] W. Dong, K. Zhou, H. Qi, C. He, J. Zhang, A tissue p system based evolutionary algorithm for multi-objective VRPTW, Swarm and Evolutionary Computation, 39 (2018) 310-322.

[4] A. Hussain, H.M. Kim, Ev prioritization and power allocation during outages: a lexicographic method-based multi-objective optimization approach, IEEE Transactions on Transportation Electrification, 7 (2021) 2474-2487.

[5] J. Zhou, L. Gao, X. Li, Ensemble of dynamic resource allocation strategies for decomposition-based multi-objective optimization, IEEE Transactions on Evolutionary Computation, 25 (2021) 710-723.

[6] W. Yankai, W. Shilong, L. Dong, S. Chunfeng, Y. Bo, An improved multi-objective whale optimization algorithm for the hybrid flow shop scheduling problem considering device dynamic reconfiguration processes, Expert Systems with Applications, 174 (2021) 114793.

[7] X. Zhang, L. Jin, C. Cui, J. Sun, A self-adaptive multi-objective dynamic differential evolution algorithm and its application in chemical engineering, Applied Soft Computing, 106 (2021) 107317.

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

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

[10] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, IEEE Transactions on Evolutionary Computation, 3 (1999) 257-271.

[11] H. Meng, S. Liu, Meng Hongyun, Liu Sanyang, ISPEA: improvement for the strength Pareto evolutionary algorithm for multiobjective optimization with immunity, in *Proceedings Fifth International Conference on Computational Intelligence and Multimedia Applications, ICCIMA 2003*, 2003, pp. 368-372.

[12] M.T. Al-Hajri, M.A. Abido, Multiobjective optimal power flow using improved strength Pareto evolutionary algorithm (spea2), in *2011 11th International Conference on Intelligent Systems Design and Applications*, 2011, pp. 1097-1103.

[13] J. Ou, J. Zheng, G. Ruan, Y. Hu, J. Zou, M. Li, S. Yang, X. Tan, A pareto-based evolutionary algorithm using decomposition and truncation for dynamic multi-objective optimization, Applied Soft Computing, 85

(2019) 105673.
[14] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on Evolutionary Computation, 11 (2007) 712-731.
[15] J. Li, J. Li, P.M. Pardalos, C. Yang, DMaOEA-εC: Decomposition-based many-objective evolutionary algorithm with the ε-constraint framework, Information Sciences, 537 (2020) 203-226.
[16] J. Luo, Y. Yang, X. Li, Q. Liu, M. Chen, K. Gao, A decomposition-based multi-objective evolutionary algorithm with quality indicator, Swarm and Evolutionary Computation. 39 (2018) 339–355.
[17] W. Lin, Q. Lin, J. Ji, Z. Zhu, C.A.C. Coello, K.-C. Wong, Decomposition-based multiobjective optimization with bicriteria assisted adaptive operator selection, Swarm and Evolutionary Computation, 60 (2021) 100790.
[18] W. Qiu, J. Zhu, G. Wu, M. Fan, P.N. Suganthan, Evolutionary many-Objective algorithm based on fractional dominance relation and improved objective space decomposition strategy, Swarm and Evolutionary Computation, 60 (2021) 100776.
[19] Y. Tian, R. Cheng, X. Zhang, F. Cheng, Y. Jin, An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility, IEEE Transactions on Evolutionary Computation, 22 (2018) 609-622.
[20] H. Chen, Y. Tian, W. Pedrycz, G. Wu, R. Wang, L. Wang, Hyperplane assisted evolutionary algorithm for many-objective optimization Problems, IEEE Transactions on Cybernetics, 50 (2020) 3367-3380.
[21] K. Shang, H. Ishibuchi, A new hypervolume-based evolutionary algorithm for many-objective optimization, IEEE Transactions on Evolutionary Computation, 24 (2020) 839-852.
[22] J. Li, G. Chen, M. Li, H. Chen, An enhanced-indicator based many-objective evolutionary algorithm with adaptive reference point, Swarm and Evolutionary Computation, 55 (2020) 100669.
[23] S. Zapotecas-Martínez, A. López-Jaimes, A. García-Nájera, LIBEA: A lebesgue indicator-based evolutionary algorithm for multi-objective optimization, Swarm and Evolutionary Computation, 44 (2019) 404-419.
[24] K. Deb, R.B. Agrawal, Simulated binary crossover for continuous search space, Complex systems, 9 (1995) 115-148.
[25] K. Deb, M. Goyal, A combined genetic adaptive search (GeneAS) for engineering design, Computer Science and informatics, 26 (1996) 30-45.
[26] Y. Wang, H. Liu, H. Long, Z. Zhang, S. Yang, Differential evolution with a new encoding mechanism for optimizing wind farm layout, IEEE Transactions on Industrial Informatics, 14 (2018) 1040-1054.
[27] I. Fister, A. Iglesias, A. Galvez, J. Del Ser, E. Osaba, I. Fister, M. Perc, M. Slavinec, Novelty search for global optimization, Applied Mathematics and Computation, 347 (2019) 865-881.
[28] L. Xiaodong, Y. Xin, Cooperatively coevolving particle swarms for large scale optimization, IEEE Transactions on Evolutionary Computation, 16 (2012) 210-224.
[29] F. Vandenbergh, A.P. Engelbrecht, A cooperative approach to particle swarm optimization, IEEE Transactions on Evolutionary Computation, 8 (2004) 225-239.

[30] M.A. Potter, The design and analysis of a computational model of cooperative coevolution, George Mason University, 1997.
[31] M.N. Omidvar, X. Li, Y. Mei, X. Yao, Cooperative co-evolution with differential grouping for large scale optimization, IEEE Transactions on Evolutionary Computation, 18 (2014) 378-393.
[32] Z. Yang, K. Tang, X. Yao, Large scale evolutionary optimization using cooperative coevolution, Information Sciences, 178 (2008) 2985-2999.
[33] R. Lan, Y. Zhu, H. Lu, Z. Liu, X. Luo, A two-phase learning-based swarm optimizer for Large-scale optimization, IEEE Transactions on Cybernetics, 51 (2021) 6284-6293.
[34] Q. Yang, W. Chen, T. Gu, H. Zhang, J.D. Deng, Y. Li, J. Zhang, Segment-based predominant learning swarm optimizer for large-scale optimization, IEEE Transactions on Cybernetics, 47 (2017) 2896-2910.
[35] Q. Yang, W. Chen, J.D. Deng, Y. Li, T. Gu, J. Zhang, A level-based learning swarm optimizer for large-scale optimization, IEEE Transactions on Evolutionary Computation, 22 (2018) 578-594.
[36] Y. Tian, X. Zheng, X. Zhang, Y. Jin, Efficient large-scale multiobjective optimization based on a competitive swarm Optimizer, IEEE Transactions on Cybernetics, 50 (2020) 3696-3708.
[37] P. Mohapatra, K. Nath Das, S. Roy, A modified competitive swarm optimizer for large scale optimization problems, Applied Soft Computing, 59 (2017) 340-362.
[38] Q. Lin, S. Liu, Q. Zhu, C. Tang, R. Song, J. Chen, C.A.C. Coello, K.-C. Wong, J. Zhang, Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems, IEEE Transactions on Evolutionary Computation, 22 (2018) 32-46.
[39] C.A. Coello Coello, M.S. Lechuga, MOPSO: a proposal for multiple objective particle swarm optimization, in: *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, 2002, vol. 2, pp. 1051-1056.
[40] M.R. Sierra, C.A. Coello Coello, Improving PSO-based multi-objective optimization using crowding, nutation and ∈-dominance, in *International conference on evolutionary multi-criterion optimization*, pp. 505-519. Springer, Berlin, Heidelberg.
[41] A.J. Nebro, J.J. Durillo, J. Garcia-Nieto, C.A. Coello Coello, F. Luna, E. Alba, SMPSO: A new PSO-based metaheuristic for multi-objective optimization, in: *2009 IEEE Symposium on Computational Intelligence in Milti-Criteria Decision-Making (MCDM)*, 2009, pp. 66-73.
[42] E.T. Oldewage, A.P. Engelbrecht, C.W. Cleghorn, The merits of velocity clamping particle swarm optimisation in high dimensional spaces, in: *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2017, pp. 1-8.
[43] H. Zille, H. Ishibuchi, S. Mostaghim, Y. Nojima, A framework for large-scale multiobjective optimization based on problem transformation, IEEE Transactions on Evolutionary Computation, 22 (2018) 260-275.
[44] C. He, R. Cheng, D. Yazdani, Adaptive offspring generation for evolutionary large-scale multiobjective optimization, IEEE Transactions on Systems, Man, and Cybernetics: Systems, (2020) 1-13.
[45] X. Zhang, Y. Tian, R. Cheng, Y. Jin, A decision variable clustering-based evolutionary algorithm for

large-scale many-objective optimization, IEEE Transactions on Evolutionary Computation, 22 (2018) 97-112.
[46] X. Ma, F. Liu, Y. Qi, X. Wang, L. Li, L. Jiao, M. Yin, M. Gong, A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables, IEEE Transactions on Evolutionary Computation, 20 (2016) 275-298.
[47] C. He, L. Li, Y. Tian, X. Zhang, R. Cheng, Y. Jin, X. Yao, Accelerating large-scale multiobjective optimization via problem reformulation, IEEE Transactions on Evolutionary Computation, 23 (2019) 949-961.
[48] R. Cheng, Y. Jin, A competitive swarm optimizer for large scale optimization, IEEE Transactions on Cybernetics, 45 (2015) 191-204.
[49] M. Li, S. Yang, X. Liu, Shift-based density estimation for Pareto-based algorithms in many-objective optimization, IEEE Transactions on Evolutionary Computation, 18 (2014) 348-365.
[50] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, IEEE Transactions on Evolutionary Computation, 20 (2016) 773-791.
[51] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, Test problems for large-scale multiobjective and many-objective optimization, IEEE Transactions on Cybernetics, 47 (2017) 4108-4121.
[52] L. Li, Q. Lin, K. Li, Z. Ming, Vertical distance-based clonal selection mechanism for the multiobjective immune algorithm, Swarm and Evolutionary Computation, 63 (2021) 100886.
[53] Y. Tian, X. Xiang, X. Zhang, R. Cheng, Y. Jin, Sampling reference points on the Pareto fronts of benchmark multi-objective optimization problems, in *Proceedings of the 2018 IEEE World Congress on Computational Intelligence (WCCI 2018)*, 2018, pp. 1-6.
[54] Y. Tian, R. Cheng, X. Zhang, Y. Jin, PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [Educational Forum], IEEE Computational Intelligence Magazine, 12 (2017) 73-87.