A Self-exploratory Competitive Swarm Optimization Algorithm for Large-Scale Multiobjective Optimization

Sheng Qi[a,b,*], Juan Zou[a,b,*], Shengxiang Yang[c], Yaochu Jin[d], Jinhua Zheng[a], Xu Yang[a,b]

[a] *Key Laboratory of Intelligent Computing and Information Processing, Ministry of Education, School of Computer Science and School of Cyberspace Science of Xiangtan University, Xiangtan, Hunan Province, China*

[b] *Faculty of School of Computer Science and School of Cyberspace Science of Xiangtan University, Xiangtan, 411105, China*

[c] *School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH,U.K.*

[d] *Department of Computer Science, University of Surrey, Guildford, Surrey GU2 7XH, U.K.*

Abstract

With the popularity of "flipped classrooms," teachers pay more attention to cultivating students' autonomous learning ability while imparting knowledge. Inspired by this, this paper proposes a Self-exploratory Competitive Swarm Optimization algorithm for Large-scale Multiobjective Optimization (SECSO). Its idea is very simple and there are no parameters that need to be adjusted. Particles evolve by exploring their neighboring space and learning from other particles in the swarm, thereby simultaneously enhancing the diversity and convergence performance of the algorithm. Compared with eight state-of-the-art large-scale multiobjective evolutionary algorithms, the proposed method exhibited outstanding performance on LSMOP problems with up to 10,000 decision variables. Unlike most existing large-scale evolutionary algorithms that usually require a large number of objective evaluations, SECSO shows the ability to find a set of well converged and diverse nondominated solutions.

Keywords: Evolutionary algorithms, Large-scale optimization, Multiobjective optimization, Self-exploratory

*Sheng Qi and Juan Zou contributed equally to this work

Email address: zoujuan@xtu.edu.cn (Juan Zou)

1. Introduction

Many real-world multiobjective optimization problems (MOPs) cannot be easily solved by mathematical programming methods [1, 2]. Due to conflicts between their objectives, finding an optimal solution to meet all the objectives and reach the optimal value is impossible. As a result, there is usually more than one optimal solution. Furthermore, it is often impossible to distinguish between good and bad solutions in the optimal sets.

To address these problems, researchers have proposed a variety of evolutionary algorithms (EAs) [3, 4]. These MOEAs show promising results in solving low-dimensional decision space problems. However, these evolutionary strategies show inferior results in the high-dimensional decision space. As the dimensionality of decision variables increases, the search space also increases exponentially. The vast search space also poses a challenge to the search efficiency of the algorithm. Generally, when the dimension of the decision variable of the problem is higher than 100, it is called a large-scale optimization problem.

Faced with such problems, researchers are studying large-scale multiobjective optimization problems (LMOPs) [5]. With the development of large-scale EAs and the demands of the era of big data, researchers are no longer limited to hundreds of decision variables, but are shifting to higher dimensions, which also poses new challenges to EAs [6]. There are a variety of solutions to large-scale multiobjective optimization problems (LMOPs), which can be roughly divided into the following three categories:

1) The first category is the co-evolution method (CC) [7], which applies the idea of divide and conquer to large-scale optimization. The CC divides many variables into several groups, and then each group is optimized independently and maintains cooperation to deal with complex problems with a large number of decision variables. In MOEA/DVA [8], variables are divided into position variables, distance variables, and mixed variables. The experimental results of this method show that compared with traditional MOEAs, MOEA/DVA exhibits a better effect in solving large-scale multiobjective optimization problems (LMOPs). However, this category of method has significant shortcomings. First, it takes much time to analyze the decision variables before grouping the decision variables. Second, when faced with complex

large-scale multiobjective optimization problems (LMOPs), decision variables are not always separable.

2) The second category is a method based on problem transformation. Zille et al. [9] used WOF to divide decision variables into multiple groups and assigned each group a weight. This method transforms the original optimization problem into a weight vector for optimization, directly achieving dimensionality reduction of decision variables. Subsequently, He et al. [10] developed the dimension reduction method by optimizing a set of weight variables and different directions in the decision space. A severe defect of MOEAs based on problem conversion is that the original problem is converted into a new problem, which will inevitably cause the loss of problem information.

3) Different from reducing the dimensionality of decision variables, some new optimizers have also appeared in the literature. These new optimizers completely break away from the CC framework. They can be regarded as the third category. LMOCSO [11] preupdates the position of the loser to change the direction of particle learning, thereby speeding up the search for optimal solutions. DGEA [12] proposes a preselection strategy to select some well-converged solutions and well-distributed solutions and adaptively uses two types of direction vectors to guide the generation of promising solutions. LMOEA-DS [6] proposes a directed sampling strategy to generate promising solutions to search. In LMOEA-DS, the reference vector set W divides the overall search space into multiple subspaces. Then, two promising search directions are determined, and the solutions are randomly sampled as guiding solutions. Finally, a complementary strategy of reference vector and elite nondominated sorting is used for environmental selection. Yang et al. [13] proposed a framework using fuzzy (FDV) decision variables to optimize large-scale multi-objective problems, which can be regarded as the fourth category. FDV preprocesses each solution before optimization, similar solutions in the population will be fuzzy. As the evolution progresses, the degree of fuzzy will be dynamically adjusted. Therefore, the search difficulty is significantly reduced. A length reduction strategy [14] is proposed to recursively shorten the length of each individual in the population to improve the performance of large-scale optimization algorithms. Liu et al. [15]

trained a specialized feedforward neural network to accelerate population convergence, the network has a three-layer model with only one hidden layer.

Research in this area is in its infancy. The vast search regions and the consequent increase of the local optimal regions reduce the search efficiency of existing EAs. Therefore, it is crucial to propose an update strategy that optimizes all variables as a whole and maintains a high degree of diversity to prevent falling into the local optima regions [16].

This paper proposes a self-exploratory competitive swarm optimization for large-scale multiobjective optimization (SECSO). The loser in the competition derivative particles in the restricted space. SECSO does not require grouping decision variables or problem conversion. In SECSO, the decision variables are optimized as a whole, improving the convergence speed of the competitive swarm optimizer (CSO) [17]. The main contributions of this paper are as follows:

1) A parameter-free particle self-exploratory strategy is proposed, which is straightforward with outstanding results. The particle self-exploratory strategy uses the loser (particles that fail in competition) as the center to explore around. Unlike CSO and its variants, the proposed SECSO does not directly abandon losers. It uses the exploration and exploitation capabilities of the loser to help the population find promising areas. The derivative radius of the particles is adaptively adjusted with the progress of evolution and the particles' fitness.

2) A large-scale multiobjective optimization algorithm SECSO is proposed. SECSO divides the entire evolution process into two stages: particle self-exploratory and competition learning. In the particle competition learning stage, the loser quickly moves closer to the winner (particles that win in competition). As the particles learning process (update the velocity and position of losers), many particles will quickly gather around the winner. The loser is discarded. This phenomenon can easily lead to the loss of population diversity. The population falls into the local optimum or has insufficient distribution. The reason is explained in Section 2. In the self-exploratory stages, the loser will derive four particles around. Different from the idea of most algorithms, particles with more excellent fitness have a larger exploration radius. As for the definition of fitness, we will elaborate on it in the Section

3. This means that particles with more excellent fitness will explore regions farther away from particles, and the exploitation of regions near the particles will be completed in the particle competition learning stage. The two stages complement each other, balancing between exploration and exploitation.

3) A cooperative strategy of searching in a restricted space and global search is proposed. Excellent particles are fundamental to guiding the direction of evolution. In the process of self-exploratory, the particle does not search the entire decision space but explores in a restricted subspace. The particle only searches in four subspaces centered around the particle. Searching in the restricted space significantly reduces the difficulty of the search and helps the population quickly search for a better solution. In the competitive learning stage, the particles that are found and that have better fitness in self-exploratory stage will guide the population to global search in the entire decision space to find more Pareto optimal solutions.

4) To verify the effectiveness of SECSO in solving LMOPs, it is compared with eight state-of-the-art MOEAs on the large-scale multiobjective test suite LSMOP [18], using problems with up to 10,000 dimensions. In addition, we compared nine state-of-the-art large-scale optimizers on the latest test suite LMF [19] to prove the ability of SECSO to solve complex LMOPs. Experimental results show that SECSO performed better under most test cases. More importantly, compared with the other MOEAs, SECSO performed better under different function evaluations.

The rest of this paper is organized as follows. We review work on large-scale optimization and summarize our motivation in Section 2. In Section 3, the details of SECSO for large-scale multiobjective optimization are presented. Section 4 presents the experimental comparisons of SECSO and eight state-of-the-art MOEAs on LMOPs. In Section 5, the conclusion is drawn.

2. Related Work And Motivation

2.1. Multiobjective Optimization Problems

MOPs can be mathematically defined as [20]:

$$\begin{cases} \min F(X) = (f_1(X), f_2(X), \ldots, f_m(X)), \\ \text{subject to } X \in \Omega, \end{cases} \tag{1}$$

where $X = (x_1, x_2, \ldots, x_D)$ is a D-dimensional decision variable vector from the decision space Ω. When D exceeds 100, it is usually called a LMOPs; $F(X)$ is an objective function vector that consists of m conflicting objective functions. A set of trade-off solutions, termed the Pareto optimal set (PS), is expected to be found for MOPs. Let X_1, $X_2 \in \Omega$; X_1 is said to dominate X_2, denoted by $X_1 \prec X_2$, if and only if $f_i(X_1) \leq f_i(X_2)$ for each $i \in \{1, \ldots, m\}$ and $f_j(X_1) < f_j(X_2)$ for at least one indicator $j \in \{1, \ldots, m\}$; if all X from Ω cannot dominate X_1, we call X_1 a nondominated or Pareto optimal solution. The set of all the Pareto optimal objective vectors is called the Pareto optimal front (PF). Any improvement in one objective of the PS is bound to deteriorate at least one other objective.

2.2. Existing Competitive Swarm Optimizer for Large-scale Optimization

To solve optimization problems, Eberhart proposed particle swarms optimization (PSO) [21], which is a process of evolution using information sharing between particles. However, when *gbest* falls into a local optimum region, all particles in the population are guided to do the same. To solve this problem, many PSO variants have been proposed [22, 23]. Liang et al. [23] proposed that PSO should not use the *gbest* update strategy but only the *pbest* to guide the particle position update. However, falling into the local optimum region is still a problem for large-scale optimization problems with this variant.

In 2015, CSO was proposed to solve large-scale optimization problems; CSO abandons *gbest* and *pbest*, it adopts pair competition among the particles in a population, replacing *gbest* and *pbest* with the winner. This competitive mechanism increases the diversity of the population in the early stages of evolution because particles learn from different particles in the same generation. The loser in the g-th generation of CSO will be updated according to the following:

$$\begin{aligned} V_l(g+1) &= r_0 V_l(g) + r_1 \left(X_w(g) - X_l(g)\right), \\ X_l(g+1) &= X_l(g) + V_l(g+1), \end{aligned} \tag{2}$$

where X_w is the position of the particle with more excellent fitness in the paired competition, and X_l and V_l are the position and initial velocity of the particle with poor fitness in the paired competition; r_0 and r_1 are the uniformly randomly distributed values in [0, 1]. Note that only the loser's speed and position are updated in the original CSO, and the winner does not change.

Some variants of CSO exist [11, 16, 24, 25, 26]. However, most of them focus on how to choose better winners. In MCSO [24], the tri-competition is used to update two-thirds of the particles in the population, and the update methods of loser1 and loser2 is the same; they learn from the same winner. In LLSO [16], particles are divided into some groups of equal size according to their fitness. The particles in the lower-level group learn from the particles in the two higher-level groups. In this way, two particles of similar fitness will not compete, further increasing convergence efficiency. In TPLSO [25], the learning process is divided into mass learning and elite learning to deal with populations in turn. The difference is that different update methods are used for the two losers in the tri-competitions in the mass learning stage. The winning particles in the competition are also updated in elite learning to obtain a better winner.

Compare to the research of CSO variants in the single-objective optimization problem, the research on MOPs is scarce. As far as we know, there are only two variants, CSOPSO [26] and LMOCSO [11]. CSOPSO uses non-dominated sorting and crowding distance to select elite particles. Unfortunately, experiments have only been conducted on a maximum of 30 decision variables. The preupdate of the particle position before the particle learning in LMOCSO has strengthens the convergence of the population. However, LMOCSO is only effective when it has sufficient computing resources. Practical real-life applications may be complex because massive function evaluations require substantial computing resources.

2.3. Motivation

A good optimizer must find a balance between fast convergence and population diversity. Enhancing the diversity of EAs to avoid falling into local optimum regions is particularly critical in large-scale optimization. On the one hand, WOF emphasizes convergence. The algorithm can converge quickly and find quasi-optimal solutions when only limited computing resources are available. However, even when computing resources are sufficient, the diversity of population is difficult to improve [11]. On the other hand, LMOCSO

has difficulty converging when the number of function evaluations is small. It has difficulty finding the optimal solutions [10]. We design SECSO to perform well in these two aspects to adapt to the environment of multiple computing resources. Because the computing resources we allocate for a project sometimes constantly change with the project's progress, the algorithm must have excellent performance under different function evaluations.

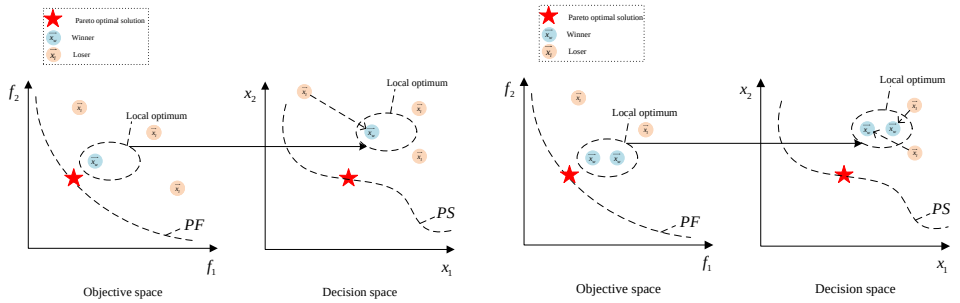

Figure 1: In the search situation of particles in the multimodal space of the g-th generation and the (g+1)-th generation in CSO, due to the winner x_w of the local optimal region, the particle x_l that failed in the competition is attracted by x_w and cannot cross the local optimum region.

The number of local optima regions usually increases in high-dimensional problems, which can easily cause the optimizer to converge prematurely. Therefore, we need the optimizer to have good convergence ability while also maintaining diversity to help the population jump out of the local optimum regions to find optimal solutions. Although CSO reduces the occurrence of local optima to a certain extent, as the decision space continues to rise, the possibility of falling into the local optima trap also increase. To better illustrate that SECSO has better exploration capabilities than existing CSO, we give examples of typical situations that occur in CSO. Let us consider the situation shown in Fig. 1(a). In the g-th iteration, there is a situation when a particle falls into the local optimum (in fact, this situation is prevalent). Since the particle $\overrightarrow{x_w}$ is in the local optimum region. It has more excellent fitness than other particles, which causes $\overrightarrow{x_w}$ to easily become a winner in the competition and attract the particles $\overrightarrow{x_l}$ that failed in the competition to move closer to the wrong local optimal region, thereby failing to find the optimal solution successfully. In fact, this kind of false attraction will

continue to accumulate. In Fig. 1(b), we demonstrate the behavior of the particles in the $(g+1)$-th iteration. Due to the guiding behavior of the winner in the g-th iteration, there are two particles in the local optimum region. They also easily win in the competition, attracting the other two losers $\overrightarrow{x_l^t}$ to move closer to this wrong region. This phenomenon will gather more winners in the local optimum region in each subsequent iteration. These winners will also attract more losers to the local optimum region (shown in Fig. 2).

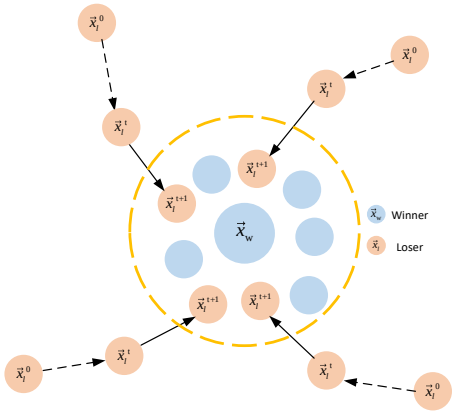

Figure 2: The state of particles after the population has evolved for multiple generations. Many particles are attracted by the winner of the local optimum region and converge to the wrong region.

To further verify our statement, we show in Fig. 3 the convergence profiles of the mean IGD values achieved by LMOCSO and SECSO on tri-objective LSMOP8 with 5,000 decision variables. Due to the lack of the particle self-exploratory mechanism, the LMOCSO quickly converges to stagnation.

Thinking carefully about this phenomenon, we found that it is caused by the LMOCSO algorithm's pursuit of convergence. We try to find inspiration from human learning to find a new and effective learning strategy. A teaching model called "flipped classroom" is popularized in pedagogy. This brand-new education model is gradually popularized in all stages of education [27, 28]. Students no longer just learn from teachers but combine independent student learning and teacher-assisted learning. This may be more reasonable for simulating the learning process of living things because living things cannot always be in a state of constantly learning from others. Likewise, particles self-exploratory before learning from other particles. Due to the self-exploratory mechanism of SECSO, the exploration of particles

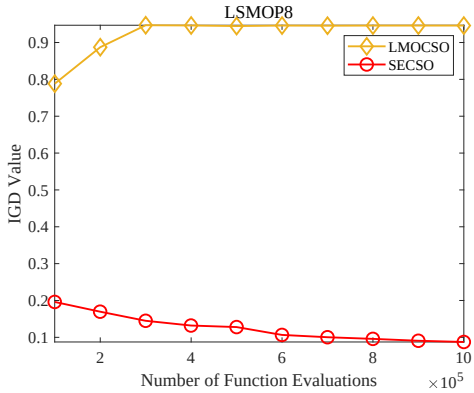

Figure 3: Convergence profiles of the mean IGD values achieved by LMOCSO and SECSO on tri-objective LSMOP8 with 5,000 decision variables

only relies on their own information. It will not be guided by any particles. The particles can easily cross the local optimum trap (Fig. 4). Furthermore, when a large number of particles gather, the self-exploratory mechanism of the particles forces the particle to explore outwards, finding other promising solutions to enhance the diversity of the population.

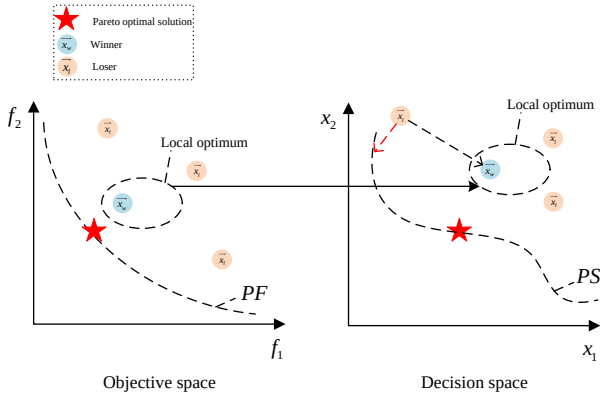

Figure 4: The self-exploratory mechanism makes the particles not attracted by the winner and skips the local optimum region to find the optimal solution.

3. PROPOSED SECSO

3.1. Particle Self-exploratory Mechanism

We adopt a displacement-based density estimation (SDE) strategy [29] to calculate the fitness of each particle. The fitness of particle p is defined as the minimum distance based on the SDE between the particle and other particles in the population, calculated as:

$$Fitness(p) = \min_{\mathbf{q}\in P\setminus\{\mathbf{p}\}} \sqrt{\sum_{i=1}^{m} \left(\max\left\{0, f_i(q) - f_i(p)\right\}\right)^2}, \tag{3}$$

where $f_i(p)$ is the i-th objective value of p, and m is the objective number. The SDE strategy can evaluate the quality of the solution based on convergence and diversity and has been widely used in many MOEAs [30, 31].

This paper proposes a particle self-exploratory mechanism, which is executed before particle competition and learning. The self-exploratory range of the particle is a circle with the particle itself as the center. Determining a circle requires a center and a radius. The radius that determines the self-exploratory range of a particle is called a derivative radius. The derivative radius is adaptively adjusted according to the fitness of the particle, and the derivative radius of the particle is calculated by the following equations:

$$w = (x_{\text{upper}} - x_{\text{lower}}) \cdot \frac{Maxgen - gen}{Maxgen}, \tag{4}$$

$$R(p) = w \cdot \frac{Fitness(p) - Fitness_{\min}}{Fitness_{\max} - Fitness_{\min}}. \tag{5}$$

$R(p)$ is the derivative radius of particle p; w is the derivative radius base of the current population; $Fitness(p)$ is the fitness of particle p; $Fitness_{\max}$ and $Fitness_{\min}$ represent the maximum fitness and minimum fitness in the current generation; x_{upper} is the upper limit of the value interval of the decision variables; x_{lower} is the lower limit of the value interval of the decision variables; $Maxgen$ is the maximum evolution generation of the algorithm; gen is the current evolution generation of the algorithm.

In the derivative formula, w decreases with an increase of evolutionary generation. Its purpose is to ensure that the algorithm has an enormous scope of exploration in the early stages of evolution and a smaller scope of

exploration in the later stages of evolution. Note that the derivative radius is positively related to the particle's fitness. A particle with excellent fitness has a large derivative radius. Generally speaking, particles with better fitness often bear the responsibility of exploitation [16]. However, this does not mean that particles with better fitness can abandon exploration, or particles with poor fitness do not need to exploit.

Due to the influence of particle competition, with the development of the evolutionary process, the particles with excellent fitness in the population are more likely to win the competition. It is easier for these particles to attract other particles to move closer to them (as shown in Fig. 2). In other words, in the particle competition stage, the learning process of the loser has primarily completed the exploitation with better fitness particles. However, the exploratory potential of particles of better fitness has not been fully utilized. In contrast, the loser is discarded after learning from the winner. The loser's potential of the exploitation space is wholly abandoned. The self-exploratory process makes up for the shortcomings in the particle competition stage at a small cost. In the self-exploratory stage, particles with poor fitness will have a smaller exploration radius, which makes the exploitation potential of the loser and the exploration potential of the winner well utilized.

Additionally, the particles with more excellent fitness gather more densely and are closer to the center of the gathering area. Therefore, particles with excellent fitness need to have a larger derivative radius to jump out of the local optimum region. In contrast, particles with poor fitness only need a small derivative radius to jump out of the local optimum region. To further illustrate the advantages of the proposed self-exploratory strategy, we assume a minimization problem $f(x_1, x_2) = -(1 - x_1/2 + x_1^5 + x_2^3) * e^{(-x_1^2 - x_2^2)}$ and the contour of the function is shown in Fig. 5. A and B are particles with different fitness levels, and a and b are the derived radii corresponding to particles A and B, respectively. The A particle has better fitness than the B particle. It is in the center of the local optimum region, so it jumps out of the local optimum region through a large derivative radius. The B particle is at the edge of the local optimum region. It jumps out of the local optimum region through a small derivative radius.

To further illustrate we show in Table 1 the IGD value of both exploration methods using 20,000 function evaluations. SECSO and its variant SECSO-1 ran independently 30 times and the mean values were recorded. In SECSO-1, the derivative radius formula is as follows:

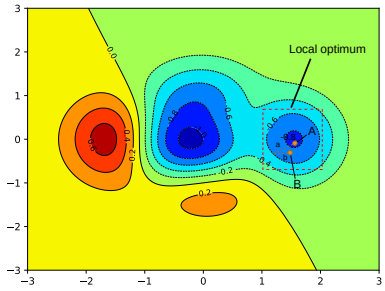

Figure 5: Illustration of a situation where the self-exploratory strategy is used to jump out of the local optimum region. Particles with better fitness need a larger derivative radius to jump out of the local optimum region.

$$w = (x_{\text{upper}} - x_{\text{lower}}) \cdot \frac{Maxgen - gen}{Maxgen}, \tag{6}$$

$$R(p) = w \cdot \frac{Fitness_{\max} - Fitness(p)}{Fitness_{\max} - Fitness_{\min}}. \tag{7}$$

where the meanings of all parameters were the same as SECSO. The only difference is that more excellent fitness particles have a smaller derivative radius. We can see the advantage of SECSO in search efficiency from Table 1.

When the number of decision variables increases, the search space increases exponentially, and the existing CSO-based algorithms do not have enough exploration ability, so the search efficiency is low [11, 32]. Our strategy limits the search space. The particles do not search the entire space, but only a limited subspace, which improves the search efficiency of the algorithm. SECSO uses particles as the center to divide the exploration direction of the particle into 2^D subspaces according to Cartesian coordinates. (i.e., when the decision variable is two, the exploration space is divided into four subspaces (+, +), (+, -), (-, +), and (-, -) according to the quadrants). After this division, each particle explores the selected subspace. Although the particle's exploration subspaces are fixed in the current generation, with the assistance of the competition stage, it is still possible for particles to search the entire decision space. The particles may be distributed at any position in

Table 1: STATISTICS OF IGD RESULTS ACHIEVED OBTAINED FROM THE TEST INSTANCES OF THE LSMOP TEST SUITE IN TWO EXPLORATION STRATEGIES. THE BEST RESULT IN EACH ROW IS HIGHLIGHTED.

Problem	N	M	D	SECSO-1	SECSO
LSMOP1	100	2	1000	4.9718e-1 (1.98e-1) −	3.9114e-1 (2.01e-1)
		3		5.6706e-1 (1.26e-1) −	3.6176e-1 (2.78e-2)
LSMOP2	100	2	1000	9.4792e-3 (7.90e-4) ≈	9.2558e-3 (1.05e-3)
		3		4.6462e-2 (3.31e-4) ≈	4.6594e-2 (4.17e-4)
LSMOP3	100	2	1000	9.8702e-1 (3.58e-2) ≈	9.9411e-1 (3.27e-2)
		3		4.8625e+0 (2.95e+0) −	9.0179e-1 (9.55e-2)
LSMOP4	100	2	1000	2.8560e-2 (1.77e-3) −	2.6759e-2 (9.81e-4)
		3		7.4454e-2 (3.12e-3) −	7.2462e-2 (2.46e-3)
LSMOP5	100	2	1000	1.2101e+0 (6.86e-1) −	1.9743e-1 (4.36e-2)
		3		9.4342e-1 (2.80e-1) −	3.2303e-1 (8.77e-3)
LSMOP6	100	2	1000	7.5810e-1 (9.88e-4) ≈	7.5810e-1 (7.65e-4)
		3		1.7692e+0 (3.52e-1) −	1.5852e+0 (4.41e-1)
LSMOP7	100	2	1000	6.1267e+0 (2.68e+0) −	2.5602e+0 (2.90e+0)
		3		9.5194e-1 (7.16e-3) ≈	9.5123e-1 (2.66e-2)
LSMOP8	100	2	1000	1.9582e+0 (7.26e-1) −	2.4001e-1 (1.75e-1)
		3		6.0196e-1 (7.13e-2) −	1.6052e-1 (2.83e-2)
LSMOP9	100	2	1000	8.0992e-1 (1.84e-4) ≈	8.0979e-1 (2.81e-4)
		3		1.2827e+0 (2.74e-1) ≈	1.2490e+0 (1.47e-1)
+/ − / ≈				0/11/7	

the decision space in the next generation. The particles in different positions have different restricted exploration regions. In this way, the search space at the particle level and the population level can balance exploration and exploitation. The particle only explores the selected subspace region at the particle level, thus significantly reducing the search space. At the population level, each particle has a different position in the decision space. Therefore, each particle has a different exploration region. As the position of particles changes, the entire decision space may be explored.

Fig. 6 illustrates the self-exploratory process in the restricted search space when there are two decision variables. Suppose that the exploration region in evolution is (-, -). The possible exploration region of particle p_1 is the dark gray region (the third quadrant with p_1 as the origin). For the entire decision space, the possible exploration region of particles is greatly reduced. However, when there is another particle p_2 in the population, the possible exploration region of particle p_2 is the light gray region (the third quadrant with p_2 as the origin). It can be seen from Fig. 6 that particle p_2 can explore the region that particle p_1 cannot.

SECSO chose a compromise plan as the exploration direction in the self-

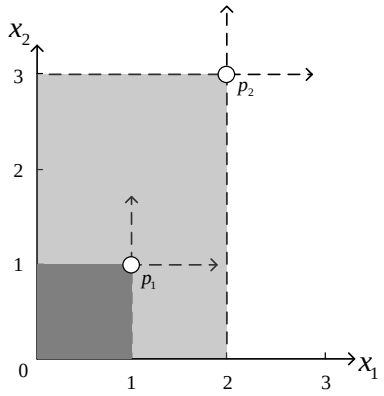

Figure 6: The self-exploratory process of particles in the restricted search space when there are two decision variables. The possible exploration region of particle p_1 is the dark gray region; the possible exploration region of particle p_2 is the gray region.

exploratory process, that is $(+1,...+D)$, $(+1,...,-(D/2+1),...-D)$, $(-1,...,+(D/2+1),...+D)$, and $(-1,...-D)$ four subspaces, where D is the number of decision variables. Take the particle p_1 in Fig. 6 as an example. In this case, there are two decision variables, and the exploration directions we choose are (+, +), (+, -), (-, +), and (-, -); four directions with particle p_1 as the origin. Therefore, in the self-exploratory stage, a particle will be derived from each of the four quadrants with p_1 as the origin. The purpose of this is to make the difference between the four subspaces as large as possible. Algorithm 1 shows the particle self-exploratory mechanism. Lines 5 to 26 can generate four derives particles. They are respectively distributed in the subspace with the original particle as the origin. Furthermore, a derivative is the generated particle in each subspace.

3.2. Algorithm Framework

The self-exploratory competition learning strategy is shown in Algorithm 2. This paper adjusted the original CSO learning strategy to make the proposed self-exploratory strategy better combined with CSO. The self-exploratory competition learning strategy divides the entire evolution process into two stages: particle self-exploratory and competition learning. Before that, all particles are sorted according to their fitness, from big to small (Lines 1-2 of Algorithm 2). The first half particle with better fitness in the current population is called P_w. The last half quantity particle with poor fitness in the current population is called P_l (Lines 4-5 in Algorithm 2). The aim is to

Algorithm 1 SelfExploration(p)

Input: p (particle).
Output: p_1, p_2, p_3, p_4 (derived particles).

$R \leftarrow R(p)$; /*Calculate according to equations (4) and (5)*/
$x \leftarrow p.x$; /*The current position of the particles*/
$z_r^k \leftarrow round(rand(0,1)), k = 1, 2, ..., D, r = 1, 2, 3, 4$; /*k is the number of decision variables and r is the number of particles*/
/*Update the first half of the decision variables of the four particles*/
for $k \leftarrow 1 : D/2$ **do**
  if $z_r^k == 1$ **then**
    $x_1^k \leftarrow x^k + R * rand(0,1)$;
    $x_2^k \leftarrow x^k - R * rand(0,1)$;
    $x_3^k \leftarrow x^k + R * rand(0,1)$;
    $x_4^k \leftarrow x^k - R * rand(0,1)$;
  end if
end for
/*Update the second half of the decision variables of the four particles*/

for $k \leftarrow D/2 + 1 : D$ **do**
  if $z_r^k == 1$ **then**
    $x_1^k \leftarrow x^k + R * rand(0,1)$;
    $x_2^k \leftarrow x^k - R * rand(0,1)$;
    $x_3^k \leftarrow x^k - R * rand(0,1)$;
    $x_4^k \leftarrow x^k + R * rand(0,1)$;
  end if
end for
/*Generate new particles*/
$p_1.x \leftarrow x_1^k$;
$p_2.x \leftarrow x_2^k$;
$p_3.x \leftarrow x_3^k$;
$p_4.x \leftarrow x_4^k$;

Algorithm 2 UpdatingParticle(P)

Input: P (current population).
Output: P' (new population).

1: Calculate the fitness of all particles in the population P according to equation (3)
2: $P \leftarrow$ Sort by fitness value from largest to smallest;
3: **while** $|P| > 1$ **do**
4: $P_w \leftarrow P(1 : end/2)$; /*Particles with better fitness */
5: $P_l \leftarrow P(end/2 + 1 : end)$; /*Particles with poor fitness*/
6: $p_w \leftarrow$ Randomly select a particle from P_w;
7: $p_l \leftarrow$ Randomly select a particle from P_l;
8: $p_1, p_2, p_3, p_4 \leftarrow SelfExploratory(p_l)$; /*Algorithmic 1*/
9: $p_w, p'_l \leftarrow LearnOperator(p_l, p_w)$; /*Equations (2)*/
10: $P' \leftarrow P' \cup \{p_w, p'_l, p_1, p_2, p_3, p_4\}$;
11: $P \leftarrow P \backslash p_w, p_l$; /*Remove p_w, p_l from the current population*/
12: **end while**

get as many of the poorer particles into the self-exploratory stage as possible to enhance the diversity of the population. p_w and p_l are randomly selected from P_w and P_l, respectively. p_l will enter the particle self-exploratory stage. After calculating the derivative radius R of particle p_l (Line 1 of Algorithm 1) according to equations 4 and 5, four derivative particles centered around p_l are generated (Lines 4-26 of Algorithm 1). During the particle's competitive learning stage, particle p_l learns from p_w (Line 9 of Algorithm 2), p_l uses its and p_w position information to update its velocity and position (equation 2). Note that the winner p_w will not learn and update. The process is repeated until all particles are selected.

As the particles learn from each other, many particles will quickly gather around the winner. The particle competitive learning mechanism ensures the population advances in the evolutionary direction. However, as the dimensions of decision space continues to rise, the particle competition mechanism appears weak. The addition of the particle self-exploratory strategy enhances the diversity of the population. It explores a broader search space so that the algorithm can find the optimal solutions faster. Even if the population is at the local optimum regions, the self-exploratory strategy can help the population have a greater chance of getting rid of the trap because the exploration radius of the particle adapts to the particle's fitness. Finally, the winner,

the updated loser, and the four derived particles enter the environmental selection together.

Algorithm 3 SECSO

Input: N (population size).
Output: P (final population).
1: $P \leftarrow InitializationPopulation(N)$;
2: $V \leftarrow InitializaitonReferenceVector(N)$; /*RVEA [33]*/
3: **while** *termination condition not* $fulfilled$ **do**
4: $P' \leftarrow UpdatingParticle(P)$; /*Algorithmic 2*/
5: $P \leftarrow EnvironmentalSelection(P \cup P', V)$; /*RVEA*/
6: **end while**

Compared with most complex grouping strategies, SECSO has a simple and easy-to-understand process. The exploration process of SECSO does not require any parameter settings, nor classification of decision variables operations, as shown in Algorithm 3. In Algorithm 3, the Line 4 is the proposed self-exploratory competitive swarm strategy. The Line 5 uses the RVEA [33] environmental selection. In addition, LMOCSO, a variant of CSO, also uses RVEA as an environment selection strategy. We use the same environment selection strategy to conduct comparative experiments to rule out the influence of other factors to prove the self-exploratory strategy's advantages better.

4. EMPIRICAL STUDIES

In this section, we report on the experiments comparing SECSO to eight state-of-the-art large-scale MOEAs (i.e., WOF, GLMO [34], MMOPSO, LMOCSO, RVEA, DGEA, LSMOEA-DS, and MOEA/D-GR [35]) on LSMOP, LMF, mDTLZ [39] and WOSGZ [40] test suite. Before 2018, ZDT [36], DTLZ [37], and F1-9 [38] test suites were commonly used to test large-scale MOPs. PS shapes in ZDT and DTLZ problems are lines or hyperplanes. Therefore, MOPs can easily obtain other solutions by searching with PS after finding a Pareto optimal solution. On the other hand, F1-F9 has a complex PS shape, which requires the population to maintain diversity while converging [35]. In 2018 LSMOP was proposed to test large-scale MOPs specifically. LMF is more complicated than LSMOP. It introduces a hybrid formulation model and designs hybrid linkages between x^p and x^d.

Additionally, to test the scalability of MOEAs, the proposed SECSO and four MOEAs (i.e., DGEA, LMOCSO, MMOPSO, MOEA/D-GR) are compared on two sets of test suites mDTLZ and WOSGZ that are scalable to any dimension. mDTLZ is a variant of DTLZ that avoids the regularly oriented PF and single distance functions in DTLZ. WOSGZ has a difficult-to-approximate PF boundary.

In the remainder of this section, we present a brief introduction to the adopted performance indicator. Then we give the parameter settings of the compared algorithms and SECSO. Each algorithm ran 20 times on large-scale benchmark MOPs independently and their mean were recorded. We verified the computational efficiency of SECSO by comparing the four MOEAs.

Then, we tested the optimization performance of the nine algorithms through comparative experiments with up to 10,000 decision variables. The Wilcoxon rank-sum test is used to compare the results at a significance level of 0.05. Symbols "+", "−" and "≈" indicate the compared algorithm is significantly better than, significantly worse than, or statistically tied by SECSO.

4.1. Performance Indicator

In the experiments, the widely used performance indicators, the inverted generational distance (IGD) [41] and HV [42], were adopted for evaluating the performance of the algorithms.

For calculating IGD, roughly 10,000 reference points are sampled on each Pareto front of LSMOP, LMF, mDTLZ and WOSGZ using [43], which is proven to be more efficient than traditional sampling. The reference solution sets in PlatEMO [44] are evenly sampled, which enables the IGD indicator to assess the qualities of the obtained sets.

4.2. Experimental Settings

4.2.1. Algorithm Parameter Settings

To make a fair comparison and ensure the algorithms performed well, we did not make any changes to the set original parameters of the compared algorithms. Specifically, for WOF, its latest version with random optimizers [34] was used, and the number of function evaluations for original problem $t1$ was set to 1000. The number of function evaluations for transformed problem $t2$ was set to 500. The number of chosen solutions to do weight optimization q was set to M+1, and the number of groups γ was set to 4. In GLMO, the optimizer uses NSGA-III, the grouping method uses ordered, the number of

variable groups is set to 4. In RVEA, the parameters controlling the rate of penalty change and the frequency of employing reference vector adaptation was set to 2 and 0.1, respectively. In DGEA, the number of reference vectors for offspring generation was set to 10, and the environmental selection operation chose the APD strategy in RVEA. In LMOEA-DS, the number of clustering referece vectors and the number of solutions to be randomly generated along each search direction was set to 10 and 30, respectively.

WOF, GLMO and MMOPSO used simulated binary crossover (SBX [45]) and polynomial variation (PM) to generate offspring. The crossover probability was set as 1; the mutation probability was set as $1/D$ (D represents the number of decision variables), and the distribution coefficient was set as 20. Finally, LMOCSO and SECSO generated offspring by using the competitive learning operator.

4.2.2. Problem Settings and Population Size

The population size of all MOEAs was set to 100. For LSMOP1–LSMOP9, the number of subcomponents in each variable group nk was set to 5. The number of objectives M was set to 2 and 3, and the number of decision variables D varied from 1,000 to 10,000.

For LMF1–LMF12, the formula model for all problems was set as a mixed model. Variable linkages for all questions were set as mixed linkages, and distance-related variables for non-uniform groupings were set. The contribution of all variables to the objective function is unbalanced. The number of objectives M was set to 3, and the number of decision variables D varied from 100 to 500.

For mDTLZ1-mDTLZ4, the number of objective M was set to 3, and the number of decision variables D was 100. For WOSGZ1-WOSGZ16, the number of decision variables D was set to 100. The objective number M of WOSGZ1-WOSGZ8 is set to 2, and the objective number M of WOSGZ9-WOSGZ16 is set to 3.

4.2.3. Termination Condition

All compared MOEAs had a consistent maximum number of function evaluations. For LSMOP problems, the maximum number of function evaluations corresponding to the dimensions of decision variables D = 1,000, D = 2,000, D = 5,000, and D=10,000 were FEs = 1,200,000, FEs = 2,500,000, FEs =5,500,000, and FEs = 10,000,000, respectively. For LMF problems, the maximum number of function evaluations was set to 2,000*D.

For mDTLZ problems, the maximum number of function evaluations was set to 20,000*D. For WOSGZ problems, the maximum number of function evaluations was set to 20,000*D.

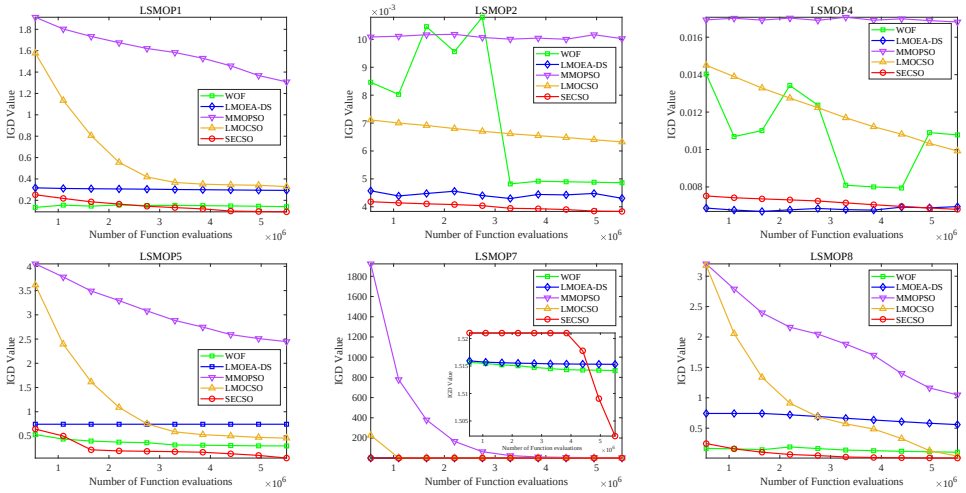

Figure 7: Convergence profiles of IGD values obtained by five algorithms on bi-objective LSMOP1, LSMOP2, LSMOP4, LSMOP5, LSMOP7 and LSMOP8 with 5000 decision variables.

4.3. Computational Efficiency

To better demonstrate the computational efficiency of the proposed SECSO, we selected four algorithms (i.e., WOF, LMOEA-DS, MMOPSO, LMOCSO) with better performance from eight compared MOEAs. In the comparison experiment of the original literature of WOF and MMOPSO, the maximum function evaluations of WOF was set to 100,000. The maximum function evaluations of LMOEA-DS was set to 80,000 in his experiments. The remaining four algorithms performed poorly in terms of convergence speed in our actual tests. The maximum value of the ordinate of the line graph increases accordingly, resulting in the figure not showing the convergence speed well. Therefore, we did not show the results of the remaining four MOEAs in the experiment. Five algorithms were compared on the different kinds of LSMOP problems, averaging over 20 runs. Among them, LSMOP1, LSMOP2, and LSMOP4 exist linear variable linkage on the PS, and LSMOP5, LSMOP7 and LSMOP8 exist non-linear variable linkage on the PS. LSMOP1 and LSMOP5 are problems with simple unimodal; LSMOP2, LSMOP4, and LSMOP8 are

mixed modalities; LSMOP7 is a problem with multimodal. In addition, LSMOP1 and LSMOP5 have a fully separable landscape; LSMOP2 has a partially separable landscape; LSMOP4, LSMOP7 and LSMOP8 have a landscape of mixed separability.

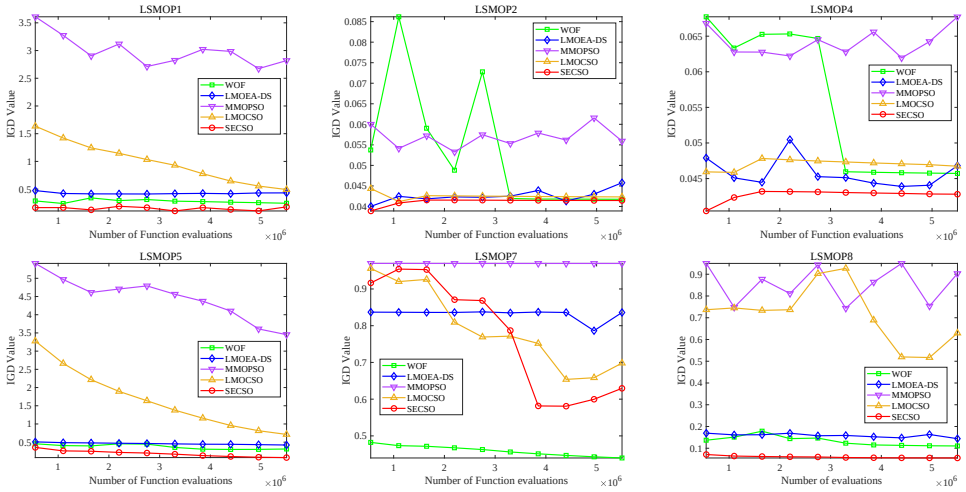

Figure 8: Convergence profiles of IGD values obtained by five algorithms on tri-objective LSMOP1, LSMOP2, LSMOP4, LSMOP5, LSMOP7 and LSMOP8 with 5000 decision variables.

Fig. 7 and Fig. 8 are the convergence profiles of IGD values obtained by five algorithms on LSMOP1, LSMOP2, LSMOP4, LSMOP5, LSMOP7 and LSMOP8 on the bi-objective LSMOP problems and tri-objective LSMOP problems, respectively. It can be seen that SECSO is better than most MOEAs with convergence advantage in 0-5,500,000 function evaluations. It means that SECSO performs better than the other four algorithms in a larger number of evaluations. We can speculate that it can adapt to environments with different computing resources. The other four algorithms converged slowly. Specifically, WOF, LMOEA-DS, LMOCSO, and SECSO can solve LSMOP1 and LSMOP5 when there are enough function evaluations. However, SECSO has a clear advantage. We speculate that the nonuniform variable groups related to the objectives in LSMOP1 and LSMOP5 caused LMOCSO to converge slowly on a certain objective. The self-exploratory method found a promising solution in the early stages of evolution, thus guiding the population's learning. In contrast, MMOPSO performed poorly. For the relatively complex problems LSMOP2 and LSMOP4, SECSO has an

overwhelming advantage over the other four algorithms. This means SECSO has potential for solving mixed modality problems. LSMOP8 is a relatively complex problem that has a mixed modality and mixed separable fitness landscape. MMOPSO and LMOCSO converged relatively poorly in the LSMOP8 because the algorithm fell into the a local optimum. Compared with the other algorithms, WOF, LMOEA-DS, and SECSO converged better. However, the computational efficiency of SECSO was significantly better than WOF and LMOEA-DS.

4.4. Optimization Performance

SECSO has significant advantages in computational efficiency and convergence speed and has an advantage in overall optimization performance. The comparative experimental results are shown in Table 2, Table 3, Table 4 and Table 5.

Table 2 shows that WOF lost information due to problem conversion. Only 6 out of 72 test cases had better performance on the bi-objective LSMOP6 and tri-objective LSMOP7. Interestingly, CSO-based variants achieved good experimental results on the most challenging LSMOP3. The GLMO algorithm has the four best performances on 72 test cases of the LSMOP problem. WOF and MMOPSO has advantages on the bi-objective LSMOP6 problem with 1,000 and 2,000 dimensional decision variables. LMOCSO shows a clear performance advantage on LSMOP9, while SECSO appears weak on this problem. We surmise this is because LSMOP9 has a disconnected PF. The self-exploratory strategy explored the quasi-optimal solution in the early stage of evolution. It prevented the population from falling into the local optimum. These quasi-optimal solutions guided the learning direction of the population and led to early convergence without finding another PF.

Table 3 shows the mean HV values of the proposed SECSO and RVEA, LMOEA-DS, DGEA, and MOEA/D-GR on the LSMOP problems. Although SECSO uses the environmental selection strategy in RVEA, it has an overwhelming advantage over RVEA. On all 72 test questions, it had better results. In particular, RVEA did not converge on the bi-objective LSMOP1, LSMOP3, LSMOP7, and LSMOP8 problems, nor on the tri-objective LSMOP3 and LSMOP6 problems. Its mean HV index was 0. SECSO solves this difficulty well. Thus, we can infer that SECSO is effective. Similarly, DGEA performed poorly on the LSMOP problems. SECSO had

Table 2: STATISTICS OF IGD RESULTS ACHIEVED BY FIVE COMPARED ALGORITHMS ON 72 TEST INSTANCES FROM LSMOP TEST SUITE. THE BEST RESULT IN EACH ROW IS HIGHLIGHTED.

Problem	M	D	WOF	GLMO	MMOPSO	LMOCSO	SECSO
LSMOP1	2	1000	8.6654e-2 (1.73e-2)−	2.3033e-1 (3.20e-3)−	4.4989e-1 (1.90e-2)−	2.7196e-2 (3.08e-2)≈	1.2244e-2 (4.06e-3)
		2000	1.0877e-1 (2.79e-2)−	2.1022e-1 (1.59e-3)−	6.8622e-1 (3.11e-2)−	1.0466e-1 (8.84e-3)−	1.3314e-2 (1.82e-3)
		5000	1.1187e-1 (1.47e-2)−	2.2616e-1 (3.15e-3)−	1.3859e+0 (1.55e-1)−	2.9549e-1 (1.44e-2)−	2.7100e-2 (6.04e-3)
		10000	1.4268e-2 (3.21e-3)≈	2.4651e-1 (3.98e-2)−	1.6663e+0 (1.06e-1)−	3.2119e-1 (1.56e-3)−	1.1730e-2 (4.37e-3)
	3	1000	1.9462e-1 (2.93e-2)−	3.3667e-1 (1.05e-2)−	2.1942e+0 (2.30e-1)−	2.7276e-1 (1.78e-2)−	1.1469e-1 (1.00e-2)
		2000	2.1000e-1 (2.76e-2)−	3.3410e-1 (2.74e-3)−	2.6958e+0 (1.27e-1)−	3.3302e-1 (1.84e-2)−	1.1814e-1 (7.50e-3)
		5000	2.5453e-1 (1.04e-2)−	3.3435e-1 (8.68e-4)−	2.9620e+0 (1.46e-1)−	5.1039e-1 (3.36e-2)−	1.4720e-1 (1.98e-2)
		10000	2.4758e-1 (4.49e-2)−	3.9875e-1 (3.88e-2)−	3.1003e+0 (2.17e-1)−	6.8515e-1 (3.46e-2)−	1.3960e-1 (2.73e-2)
LSMOP2	2	1000	9.5342e-3 (2.95e-3)−	8.0848e-3 (1.34e-3)−	2.8826e-2 (4.83e-4)−	1.9770e-2 (5.45e-4)−	4.8533e-3 (2.78e-4)
		2000	6.2771e-3 (6.44e-4)−	5.2039e-3 (2.76e-5)−	1.8116e-2 (1.95e-4)−	1.1110e-2 (1.52e-4)−	4.1962e-3 (1.95e-4)
		5000	4.7780e-3 (2.59e-4)−	3.9924e-3 (6.15e-3)−	1.0214e-2 (1.63e-4)−	6.2571e-3 (3.91e-5)−	3.7529e-3 (3.91e-5)
		10000	5.1773e-3 (8.37e-3)−	4.9924e-3 (5.58e-3)−	7.1863e-3 (2.38e-4)−	4.7115e-3 (7.55e-6)−	3.7537e-3 (3.56e-6)
	3	1000	4.5820e-2 (3.67e-4)−	4.4518e-2 (7.91e-5)−	6.8763e-2 (1.95e-3)−	4.7526e-2 (3.64e-4)−	4.3979e-2 (5.62e-4)
		2000	4.2821e-2 (1.77e-4)−	4.4005e-2 (7.95e-4)−	5.9181e-2 (1.94e-3)−	4.3880e-2 (6.82e-5)−	4.2261e-2 (1.08e-4)
		5000	4.1683e-2 (5.85e-3)−	4.4402e-2 (3.15e-4)−	5.8010e-2 (1.36e-3)−	4.2203e-2 (8.28e-5)−	4.1538e-2 (5.44e-5)
		10000	4.3135e-2 (6.05e-3)−	4.4987e-2 (3.88e-4)−	5.6975e-2 (4.99e-4)−	4.1721e-2 (4.09e-5)−	4.1257e-2 (1.07e-5)
LSMOP3	2	1000	1.2134e+0 (5.14e-1)−	1.5762e+0 (4.19e-5)−	2.0545e+1 (2.98e+0)−	7.0721e-1 (1.21e-5) ≈	7.0817e-1 (1.68e-4)
		2000	1.0808e+0 (5.61e-1)−	1.5796e+0 (6.33e-5)−	2.4182e+1 (9.89e-1)−	7.0720e-1 (4.93e-6) ≈	7.0860e-1 (4.71e-4)
		5000	1.1730e+0 (5.74e-1)−	1.5817e+0 (6.15e-4)−	2.6490e+1 (1.05e+0)−	7.0725e-1 (5.53e-6) ≈	7.0955e-1 (1.43e-3)
		10000	1.1513e+0 (1.08e-1)−	1.7887e+0 (6.88e-5)−	2.6758e+1 (0.00e+0)−	7.0728e-1 (1.85e-5) ≈	7.1141e-1 (4.46e-4)
	3	1000	2.1882e+0 (1.67e+0)−	8.5082e-1 (6.46e-5)≈	1.1247e+1 (4.55e-1)−	2.6823e+0 (1.52e+0)−	8.5072e-1 (4.31e-1)
		2000	8.8649e-1 (5.54e-2)≈	8.6072e-1 (6.15e-5)≈	1.4296e+1 (1.74e+0)−	3.0868e+0 (2.40e+0)−	8.8134e-1 (2.44e-2)
		5000	1.0785e+0 (3.08e-1)−	8.6076e-1 (8.15e-5)≈	1.3965e+1 (6.48e-1)−	2.9994e+0 (2.09e+0)−	8.6072e-1 (3.34e-1)
		10000	1.0220e+0 (7.37e-1)−	8.5082e-1 (6.46e-5)≈	1.2611e+1 (7.86e-1)−	2.5263e+0 (1.86e+0)≈	8.4108e-1 (6.40e-2)
LSMOP4	2	1000	2.3101e-2 (2.73e-3)−	2.1469e-2 (5.80e-5)−	5.8266e-2 (1.57e-3)−	2.4926e-2 (1.59e-3)−	7.2973e-3 (5.05e-4)
		2000	1.4801e-2 (3.10e-3)−	1.1825e-2 (8.02e-6)−	3.4004e-2 (2.75e-4)−	1.7662e-2 (6.28e-4)−	5.4532e-3 (5.54e-4)
		5000	1.0293e-2 (5.03e-3)−	6.2603e-3 (8.15e-4)−	1.6737e-2 (1.50e-4)−	9.6555e-3 (1.79e-4)−	4.5876e-3 (3.28e-4)
		10000	1.1598e-2 (3.27e-4)−	6.9878e-3 (5.48e-4)−	1.0606e-2 (2.14e-4)−	6.4973e-3 (2.45e-6)−	4.7344e-3 (1.45e-4)
	3	1000	7.3575e-2 (3.48e-3)−	6.3744e-2 (9.24e-4)−	1.2860e-1 (2.10e-3)−	8.3904e-2 (1.22e-3)−	5.6664e-2 (1.40e-3)
		2000	5.8397e-2 (2.60e-4)−	4.9561e-2 (1.46e-4)−	8.5706e-2 (2.21e-3)−	6.0111e-2 (8.42e-4)−	4.7253e-2 (8.52e-4)
		5000	4.5776e-2 (7.61e-5)−	4.3102e-2 (1.15e-4)−	6.3808e-2 (2.09e-3)−	4.6706e-2 (1.57e-4)−	4.2941e-2 (1.66e-4)
		10000	5.4344e-2 (4.78e-4)−	6.9878e-2 (5.87e-4)−	6.0249e-2 (1.02e-3)−	4.3358e-2 (6.09e-5)−	4.1711e-2 (6.57e-5)
LSMOP5	2	1000	2.7985e-1 (1.32e-1)−	3.6395e-1 (6.03e-3)−	6.9529e-1 (1.04e-1)−	1.4674e-2 (1.55e-3)−	9.6902e-3 (1.37e-3)
		2000	2.9459e-1 (9.74e-2)−	3.7479e-1 (2.06e-3)−	1.3412e+0 (5.37e-2)−	1.0055e-1 (1.04e-1)−	1.3090e-2 (8.86e-4)
		5000	2.4032e-1 (1.52e-1)−	3.2149e-1 (3.78e-3)−	2.4942e+0 (9.93e-2)−	3.3481e-1 (6.01e-3)−	2.1787e-2 (6.29e-4)
		10000	1.5410e-1 (1.99e-2)−	3.4678e-1 (4.47e-3)−	3.1166e+0 (3.07e-3)−	3.9010e-1 (7.19e-3)−	2.9252e-2 (3.32e-3)
	3	1000	3.1470e-1 (9.51e-2)−	3.6744e-1 (4.31e-3)−	6.7214e-1 (1.02e-1)−	5.1919e-1 (3.60e-2)−	7.1116e-2 (6.05e-3)
		2000	2.0205e-1 (1.91e-2)−	3.1264e-1 (1.11e-2)−	8.2739e-1 (7.95e-2)−	5.3620e-1 (1.95e-2)−	7.2101e-2 (4.12e-3)
		5000	3.1238e-1 (4.87e-2)−	3.3173e-1 (1.15e-2)−	4.3621e+0 (6.02e-1)−	7.2490e-1 (1.87e-2)−	7.5106e-2 (3.08e-3)
		10000	3.1361e-1 (7.55e-2)−	3.9878e-1 (3.48e-3)−	4.9395e+0 (4.26e-1)−	8.5408e-1 (7.86e-2)−	8.1784e-2 (1.72e-3)
LSMOP6	2	1000	2.1743e-1 (7.28e-2) +	2.1064e-1 (2.13e-2)+	3.3680e-1 (5.75e-2)+	6.9762e-1 (1.08e-1)≈	7.5294e-1 (1.51e-3)
		2000	1.7360e-1 (8.61e-3) +	2.9369e-1 (8.41e-2)+	3.5352e-1 (4.01e-2)+	7.4972e-1 (6.57e-4)−	7.1442e-1 (3.94e-2)
		5000	1.7144e-1 (1.55e-2) +	2.8977e-1 (8.87e-2)+	7.4727e-1 (4.06e-4)−	7.4681e-1 (6.25e-4)−	6.4916e-1 (9.59e-3)
		10000	2.8370e-1 (1.36e-1) +	2.8676e-1 (1.74e-2)+	7.4471e-1 (8.33e-5)−	7.4439e-1 (1.34e-5)−	3.8453e-1 (5.91e-3)
	3	1000	1.3282e+0 (4.90e-2)−	1.3095e+0 (1.06e-3)−	1.4390e+1 (2.67e+1)−	1.7167e+0 (8.79e-2)−	1.2553e+0 (7.30e-2)
		2000	1.4256e+0 (2.09e-2)−	1.3182e+0 (1.74e-4)≈	1.4136e+1 (2.60e+1)≈	1.7488e+0 (6.02e-1)≈	1.3141e+0 (3.93e-3)
		5000	1.3988e+0 (5.48e-3)−	8.9718e-1 (6.15e-1)+	1.7700e+0 (5.59e-2)−	2.2556e+0 (5.31e-1)−	1.3228e+0 (4.98e-3)
		10000	1.4095e+0 (3.48e-3)−	3.9487e+3 (2.30e+2)−	1.3759e+0 (4.04e-1)≈	3.8308e+0 (8.68e-1)−	1.3733e+0 (1.48e-1)
LSMOP7	2	1000	1.6188e+0 (1.50e-1)≈	1.5117e+0 (1.07e-4)≈	3.0143e+0 (4.50e-1)−	1.8989e+0 (3.28e-1)≈	1.5748e+0 (1.38e-1)
		2000	1.6609e+0 (1.85e-1)≈	1.7148e+0 (2.01e-5)≈	3.6069e+0 (1.37e+0)−	1.8150e+0 (1.83e-1)≈	1.6229e+0 (3.72e-1)
		5000	1.5141e+0 (1.62e-4)≈	1.7487e+0 (3.54e-5)≈	4.7946e+0 (3.98e-1)−	1.7643e+0 (2.41e-1)−	1.5113e+0 (7.43e-3)
		10000	1.6481e+0 (9.57e-3)≈	1.7453e+0 (7.51e-5)≈	1.0788e+2 (3.25e+0)−	1.6345e+0 (1.37e-1)≈	1.5221e+0 (1.32e-3)
	3	1000	8.0461e-1 (5.82e-2)≈	8.4857e-1 (3.04e-4)−	1.0815e+0 (1.74e-3)−	9.4594e-1 (2.02e-6)≈	7.8069e-1 (1.95e-1)
		2000	6.6166e-1 (2.85e-1)≈	8.4060e-1 (7.57e-5)−	1.0098e+0 (1.04e-3)−	8.2600e-1 (1.56e-1)≈	6.5631e-1 (1.93e-1)
		5000	4.3989e-1 (8.73e-3) +	8.3643e-1 (8.48e-5)−	9.6982e-1 (1.49e-4)−	7.4481e-1 (1.66e-1)≈	6.8635e-1 (1.74e-1)
		10000	5.8316e-1 (1.00e-1) +	8.8794e-1 (8.48e-4)−	9.5742e-1 (7.51e-5)−	7.1462e-1 (1.55e-1)≈	6.8408e-1 (1.20e-2)
LSMOP8	2	1000	9.1184e-2 (3.07e-3)−	2.9012e-1 (9.14e-4)−	2.7795e-1 (6.23e-3)−	2.4601e-2 (1.05e-3)−	8.4323e-3 (5.14e-4)
		2000	1.0683e-1 (4.08e-3)−	3.1089e-1 (4.69e-3)−	3.8643e-1 (2.23e-2)−	2.3823e-2 (1.52e-3)−	7.2031e-3 (2.60e-4)
		5000	9.5875e-2 (9.54e-3)−	4.0099e-1 (3.88e-3)−	1.2215e+0 (1.10e-1)−	3.7056e-2 (2.82e-3)−	8.3166e-3 (1.31e-4)
		10000	1.5040e-1 (5.76e-3)−	4.5634e-1 (7.87e-3)−	2.0708e+0 (0.00e+0)−	3.4942e-1 (3.00e-2)−	1.0423e-2 (1.27e-3)
	3	1000	1.3903e-1 (3.90e-2)−	1.4175e-1 (1.52e-3)−	3.3475e-1 (2.13e-1)−	5.1078e-1 (3.05e-2)−	5.7117e-2 (1.92e-4)
		2000	1.8090e-1 (4.13e-3)−	1.3463e-1 (4.15e-3)−	7.3598e-1 (1.11e-1)−	5.2044e-1 (1.43e-2)−	5.6392e-2 (7.82e-4)
		5000	1.1055e-1 (2.58e-3)−	1.1564e-1 (3.58e-3)−	8.5002e-1 (5.52e-2)−	6.2968e-1 (2.10e-1)−	5.5933e-2 (2.90e-4)
		10000	1.3934e-1 (2.06e-2)−	1.4355e-1 (6.48e-3)−	8.8212e-1 (7.57e-2)−	5.1437e-1 (3.06e-3)−	5.5233e-2 (1.63e-4)
LSMOP9	2	1000	4.6400e-1 (3.22e-3)≈	5.1481e-1 (1.96e-2)+	1.2518e-1 (1.57e-2)+	9.2290e-2 (8.69e-4) +	8.0936e-1 (4.33e-5)
		2000	3.4401e-1 (2.18e-1)+	4.7507e-1 (3.55e-3)+	1.7664e+0 (4.97e-1)−	4.8950e-2 (1.62e-3) +	8.0740e-1 (2.00e-3)
		5000	3.3363e-1 (3.05e-1)+	5.5598e-1 (5.48e-2)+	3.3287e+0 (1.97e-1)−	2.7359e-2 (8.22e-4) +	8.0742e-1 (1.47e-4)
		10000	2.4441e-2 (1.98e-3) +	5.7895e-1 (5.78e-2)+	4.1717e+0 (0.00e+0)−	2.6114e-1 (1.19e-1)+	8.0718e-1 (7.93e-6)
	3	1000	1.3336e+0 (1.48e-1)−	1.1588e+0 (1.94e-2)−	1.4082e+1 (2.13e+0)−	6.2816e-1 (8.69e-2) ≈	7.0098e-1 (2.72e-1)
		2000	1.5377e+0 (3.03e-4)≈	1.1509e+0 (8.37e-3)−	1.9228e+1 (1.33e+0)−	6.7192e-1 (1.48e-2) ≈	7.3455e-1 (2.65e-1)
		5000	1.4311e+0 (2.12e-2)−	1.1454e+0 (2.15e-2)−	2.5509e+1 (2.32e+0)−	6.8762e-1 (1.35e-2) ≈	7.9086e-1 (2.77e-1)
		10000	1.1437e+0 (4.35e-4)−	1.1487e+0 (9.84e-2)−	2.7710e+1 (1.05e+0)−	6.5761e-1 (8.12e-2) ≈	1.0094e+0 (2.75e-1)
+/ − / ≈			9/53/10	9/54/9	3/67/2	4/49/19	

Table 3: STATISTICS OF HV RESULTS OBTAINED BY FIVE COMPARED ALGORITHMS ON 72 TEST INSTANCES FROM LSMOP TEST SUITE. THE BEST RESULT IN EACH ROW IS HIGHLIGHTED.

Problem	M	D	RVEA	LMOEA-DS	DGEA	MOEA/D-GR	SECSO
LSMOP1	2	1000	0.0000e+0 (0.00e+0)−	2.4496e-1 (5.09e-3)−	0.0000e+0 (0.00e+0)−	2.8148e-1 (2.61e-3)−	5.6929e-1 (5.38e-3)
		2000	0.0000e+0 (0.00e+0)−	2.4848e-1 (5.74e-3)−	0.0000e+0 (0.00e+0)−	2.6203e-1 (2.63e-3)−	5.6801e-1 (2.33e-3)
		5000	0.0000e+0 (0.00e+0)−	2.4670e-1 (1.05e-3)−	0.0000e+0 (0.00e+0)−	2.1344e-1 (1.32e-3)−	5.5480e-1 (5.79e-4)
		10000	0.0000e+0 (0.00e+0)−	2.5026e-1 (1.35e-5)−	0.0000e+0 (0.00e+0)−	2.1568e-1 (6.15 e-3)−	4.4664e-1 (8.37e-4)
	3	1000	0.0000e+0 (0.00e+0)−	3.4504e-1 (2.93e-3)−	0.0000e+0 (0.00e+0)−	5.8321e-5 (8.25e-5)−	7.2127e-1 (1.67e-2)
		2000	1.7299e-1 (1.64e-1)−	3.5556e-1 (3.92e-3)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	7.2281e-1 (8.86e-3)
		5000	1.1739e-1 (6.80e-3)−	3.4956e-1 (3.99e-3)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	6.7537e-1 (3.51e-2)
		10000	1.3622e-1 (5.58e-2)−	3.4673e-1 (6.22e-3)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	7.0929e-1 (4.60e-2)
LSMOP2	2	1000	5.4207e-1 (9.33e-4)−	5.7378e-1 (2.84e-4)−	5.2785e-1 (4.14e-4)−	5.5375e-1 (8.64e-4)−	5.7928e-1 (4.70e-4)
		2000	5.6059e-1 (2.90e-4)−	5.7768e-1 (4.09e-4)−	5.5215e-1 (3.81e-4)−	5.6758e-1 (3.51e-4)−	5.8051e-1 (4.16e-4)
		5000	5.7274e-1 (8.99e-5)−	5.7987e-1 (1.54e-4)−	5.6575e-1 (3.39e-4)−	5.7566e-1 (7.08e-5)−	5.8161e-1 (1.74e-4)
		10000	5.7704e-1 (8.97e-5)−	5.8053e-1 (4.69e-4)≈	5.7022e-1 (1.22e-3)−	5.4687e-1 (3.15e-4)−	5.8164e-1 (9.78e-6)
	3	1000	8.2219e-1 (4.79e-4)−	8.0370e-1 (6.18e-3)−	7.7618e-1 (7.82e-3)−	7.7627e-1 (1.15e-3)−	8.3284e-1 (1.25e-3)
		2000	8.3114e-1 (3.77e-5)−	8.0334e-1 (3.73e-3)−	7.8505e-1 (7.88e-3)−	7.8869e-1 (2.45e-4)−	8.3698e-1 (3.43e-4)
		5000	8.3689e-1 (4.53e-6)−	8.0711e-1 (5.40e-3)−	7.9329e-1 (6.03e-3)−	7.9598e-1 (6.89e-3)−	8.3934e-1 (6.58e-5)
		10000	8.3829e-1 (4.20e-5)−	8.1088e-1 (6.87e-3)−	8.0677e-1 (2.66e-3)−	7.1698e-1 (6.78e-3)−	8.4078e-1 (2.75e-5)
LSMOP3	2	1000	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	9.0548e-2 (1.54e-4)≈	8.9544e-2 (2.15e-4)
		2000	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	8.9766e-2 (1.30e-3)≈	8.9000e-2 (6.02e-4)
		5000	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	7.4693e-2 (1.70e-2)−	8.7788e-2 (1.85e-3)
		10000	4.5405e-2 (6.42e-2)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	7.1588e-2 (3.48e-2)−	8.5419e-2 (5.65e-4)
	3	1000	0.0000e+0 (0.00e+0)−	9.1057e-2 (8.32e-5) +	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	9.0909e-2 (3.12e-2)
		2000	0.0000e+0 (0.00e+0)−	9.1093e-2 (1.56e-4) +	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	7.0648e-2 (3.51e-2)
		5000	0.0000e+0 (0.00e+0)−	9.0920e-2 (1.59e-5) ≈	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	9.0909e-2 (4.12e-2)
		10000	0.0000e+0 (0.00e+0)−	9.1007e-2 (1.21e-5) +	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	5.2274e-2 (5.46e-2)
LSMOP4	2	1000	5.1923e-1 (7.77e-4)−	5.5555e-1 (2.67e-4)−	4.8084e-1 (6.70e-4)−	5.3236e-1 (5.05e-3)−	5.7545e-1 (7.45e-4)
		2000	5.4654e-1 (5.27e-4)−	5.6823e-1 (3.38e-4)−	5.2249e-1 (2.04e-3)−	5.4850e-1 (2.11e-4)−	5.7849e-1 (9.63e-4)
		5000	5.6509e-1 (3.21e-4)−	5.7556e-1 (2.18e-4)≈	5.5218e-1 (1.00e-3)−	5.6625e-1 (1.19e-4)−	5.7994e-1 (7.01e-4)
		10000	5.7178e-1 (4.94e-5)−	5.7844e-1 (1.93e-4)−	5.6069e-1 (1.51e-3)−	5.5468e-1 (3.19e-4)−	5.7982e-1 (2.46e-4)
	3	1000	7.8015e-1 (7.79e-4)−	7.7844e-1 (1.35e-3)−	6.8052e-1 (7.00e-3)−	7.1851e-1 (5.64e-4)−	8.1356e-1 (1.82e-3)
		2000	8.0842e-1 (1.10e-3)−	7.9123e-1 (6.83e-3)−	7.4640e-1 (9.72e-3)−	7.5408e-1 (8.48e-4)−	8.2586e-1 (1.69e-4)
		5000	8.2686e-1 (2.17e-4)−	8.0537e-1 (7.04e-4)−	7.7479e-1 (3.41e-3)−	7.8051e-1 (6.15e-4)−	8.3525e-1 (2.29e-4)
		10000	8.3277e-1 (1.09e-3)−	8.0754e-1 (4.51e-3)−	7.9091e-1 (8.15e-3)−	7.4897e-1 (6.45e-4)−	8.3868e-1 (2.95e-4)
LSMOP5	2	1000	0.0000e+0 (0.00e+0)−	9.0909e-2 (1.87e-4)−	0.0000e+0 (0.00e+0)−	1.1296e-1 (2.47e-3)−	3.3556e-1 (2.36e-3)
		2000	0.0000e+0 (0.00e+0)−	9.0909e-2 (2.14e-4)−	0.0000e+0 (0.00e+0)−	7.5041e-2 (1.28e-2)−	3.2996e-1 (1.48e-3)
		5000	0.0000e+0 (0.00e+0)−	9.0909e-2 (2.87e-4)−	0.0000e+0 (0.00e+0)−	3.4649e-4 (3.48e-4)−	3.1650e-1 (1.05e-3)
		10000	0.0000e+0 (0.00e+0)−	9.0909e-2 (3.25e-8)−	0.0000e+0 (0.00e+0)−	3.9788e-4 (5.48e-4)−	3.0727e-1 (4.36e-3)
	3	1000	9.0904e-2 (6.68e-7)−	3.4300e-1 (5.74e-4)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	5.0885e-1 (1.13e-2)
		2000	9.0904e-2 (4.61e-7)−	3.4212e-1 (1.87e-4)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	5.0004e-1 (3.13e-3)
		5000	9.0901e-2 (5.41e-8)−	3.4189e-1 (1.19e-3)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	4.9360e-1 (3.57e-3)
		10000	9.0899e-2 (5.87e-7)−	3.4087e-1 (2.48e-3)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	4.8490e-1 (4.35e-3)
LSMOP6	2	1000	1.7013e-2 (1.97e-2)−	8.3649e-2 (1.63e-2) +	0.0000e+0 (0.00e+0)−	8.5273e-2 (9.38e-5)+	6.7573e-2 (3.00e-3)
		2000	1.5029e-2 (3.01e-2)−	9.6679e-2 (2.85e-3) +	0.0000e+0 (0.00e+0)−	8.2031e-2 (4.51e-5)+	7.6515e-2 (1.31e-4)
		5000	7.9222e-2 (1.47e-3)−	1.1219e-1 (2.13e-3) +	0.0000e+0 (0.00e+0)−	8.5296e-2 (2.14e-3)+	8.4585e-2 (3.40e-4)
		10000	4.2271e-2 (5.98e-2)−	1.2272e-1 (2.57e-3) +	0.0000e+0 (0.00e+0)−	8.1984e-2 (1.48e-3)≈	9.0314e-2 (3.69e-3)
	3	1000	0.0000e+0 (0.00e+0)≈	1.0125e-2 (3.60e-3) +	0.0000e+0 (0.00e+0)≈	0.0000e+0 (0.00e+0)≈	0.0000e+0 (0.00e+0)
		2000	0.0000e+0 (0.00e+0)≈	1.0683e-2 (3.94e-4) +	0.0000e+0 (0.00e+0)≈	0.0000e+0 (0.00e+0)≈	0.0000e+0 (0.00e+0)
		5000	0.0000e+0 (0.00e+0)≈	1.2224e-2 (1.24e-3) +	0.0000e+0 (0.00e+0)≈	0.0000e+0 (0.00e+0)≈	0.0000e+0 (0.00e+0)
		10000	0.0000e+0 (0.00e+0)≈	4.8047e-3 (3.18e-3) +	0.0000e+0 (0.00e+0)≈	0.0000e+0 (0.00e+0)≈	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)≈	0.0000e+0 (0.00e+0)≈	0.0000e+0 (0.00e+0)
		2000	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)
		5000	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)
		10000	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)
	3	1000	9.0885e-3 (6.68e-3)−	7.9749e-4 (1.59e-3)−	0.0000e+0 (0.00e+0)−	9.0796e-2 (2.54e-5)≈	8.8965e-2 (2.67e-3)
		2000	1.0037e-2 (9.69e-4)−	4.0562e-2 (5.75e-3)−	0.0000e+0 (0.00e+0)−	9.0845e-2 (1.14e-5)≈	9.1677e-2 (1.17e-3)
		5000	1.5169e-2 (1.14e-3)−	7.4281e-2 (1.55e-3)≈	0.0000e+0 (0.00e+0)−	9.0877e-2 (2.15e-4)≈	9.1776e-2 (7.41e-4)
		10000	6.1592e-2 (4.14e-2)−	7.9188e-2 (5.20e-3)≈	0.0000e+0 (0.00e+0)−	9.0785e-2 (3.15e-4)≈	9.0815e-2 (3.84e-4)
LSMOP8	2	1000	0.0000e+0 (0.00e+0)−	1.2148e-1 (2.45e-3)−	0.0000e+0 (0.00e+0)−	3.2303e-1 (5.00e-4)−	3.3752e-1 (8.25e-4)
		2000	0.0000e+0 (0.00e+0)−	1.0321e-1 (3.72e-3)−	0.0000e+0 (0.00e+0)−	3.2623e-1 (3.05e-4)−	3.3992e-1 (4.13e-4)
		5000	0.0000e+0 (0.00e+0)−	9.0909e-2 (6.05e-3)−	0.0000e+0 (0.00e+0)−	3.2656e-1 (3.45e-3)−	3.3888e-1 (2.03e-4)
		10000	0.0000e+0 (0.00e+0)−	9.0909e-2 (2.74e-10)−	0.0000e+0 (0.00e+0)−	3.2555e-1 (3.45e-4)−	3.3684e-1 (2.03e-3)
	3	1000	7.1299e-2 (1.27e-4)−	3.9357e-1 (6.83e-3)−	2.1812e-2 (3.14e-2)−	9.4139e-2 (9.13e-3)−	5.4257e-1 (6.95e-4)
		2000	7.8800e-2 (1.47e-5)−	4.0181e-1 (7.45e-3)−	5.7258e-2 (2.75e-2)−	1.0195e-1 (8.93e-3)−	5.4805e-1 (6.15e-4)
		5000	8.3234e-2 (6.70e-5)−	4.1281e-1 (1.53e-2)−	4.2017e-2 (2.38e-3)−	1.0987e-1 (1.15e-3)−	5.4716e-1 (1.94e-3)
		10000	8.7828e-2 (4.35e-3)−	4.0999e-1 (8.29e-3)−	5.6106e-2 (4.61e-3)−	1.9887e-1 (2.15e-3)−	5.5045e-1 (6.76e-4)
LSMOP9	2	1000	3.3040e-2 (6.69e-3)−	1.5421e-1 (1.63e-2) +	0.0000e+0 (0.00e+0)−	2.0237e-1 (2.30e-4)+	9.0959e-2 (4.28e-4)
		2000	1.5423e-2 (1.03e-2)−	1.7966e-1 (7.59e-3) +	0.0000e+0 (0.00e+0)−	2.2017e-1 (2.00e-4)+	9.1184e-2 (2.15e-4)
		5000	2.0934e-2 (2.96e-2)−	1.8354e-1 (1.20e-2) ≈	0.0000e+0 (0.00e+0)−	2.2972e-1 (2.15e-4)+	9.1208e-2 (1.80e-5)
		10000	2.7917e-2 (3.95e-2)−	1.9819e-1 (2.47e-3) +	0.0000e+0 (0.00e+0)−	2.4566e-1 (1.12e-4)+	9.1303e-2 (1.64e-6)
	3	1000	1.4503e-2 (1.17e-2)−	1.9035e-1 (1.52e-4) +	0.0000e+0 (0.00e+0)−	8.2270e-2 (5.48e-5)−	1.5647e-1 (1.90e-2)
		2000	1.1041e-2 (1.08e-3)−	1.9119e-1 (2.41e-4) +	0.0000e+0 (0.00e+0)−	8.5285e-2 (2.72e-3)−	1.8523e-1 (4.89e-5)
		5000	1.2571e-2 (1.02e-3)−	1.9070e-1 (6.27e-4) +	0.0000e+0 (0.00e+0)−	8.1914e-2 (3.15e-1)−	1.6611e-1 (2.70e-2)
		10000	7.0112e-3 (2.79e-3)−	1.9094e-1 (1.73e-4) ≈	0.0000e+0 (0.00e+0)−	8.1634-2 (1.99e-1)−	1.6599e-1 (2.71e-2)
+/ − / ≈			0/64/8	17/44/11	0/64/8	7/50/15	

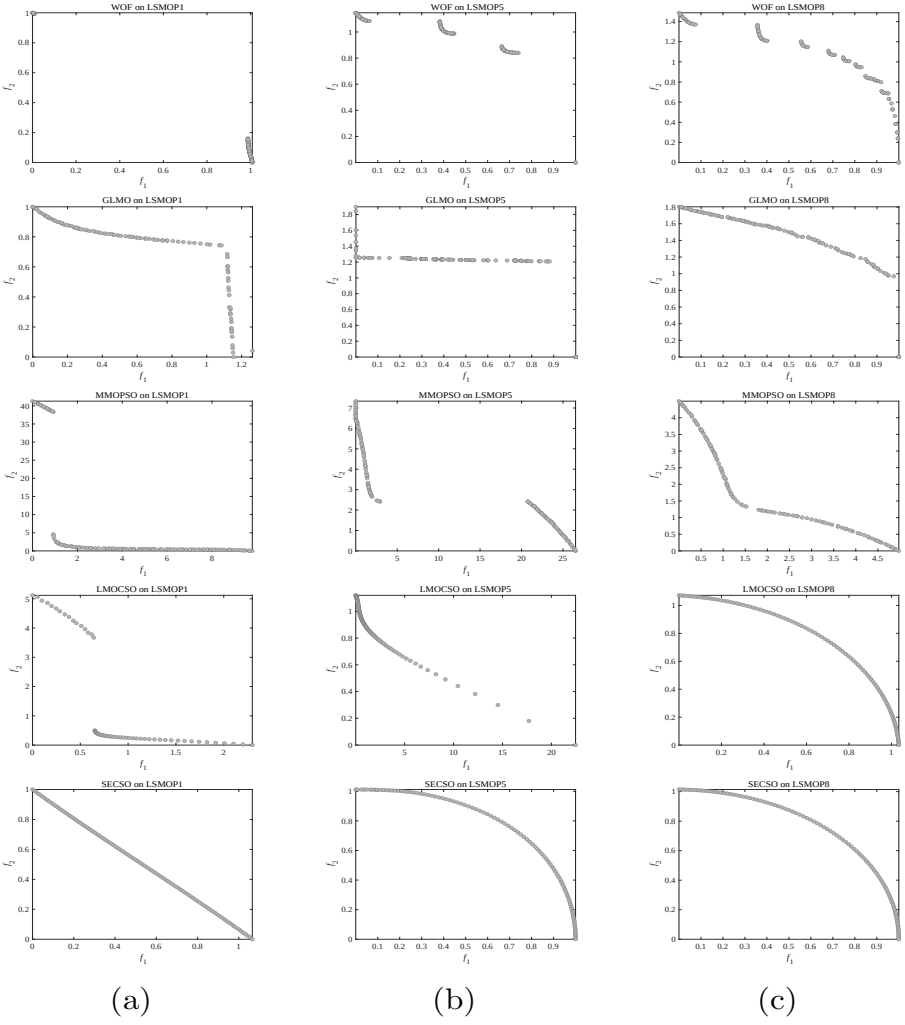

Figure 9: Nondominated solutions with the median IGD values obtained by WOF, MOEA/DVA, MMOPSO, LMOCSO and SECSO on bi-objective LSMOP1, LSMOP5 and LSMOP8 with 5,000 decision variables.

better results in 49 out of 72 test questions. We speculate that the possible reason is that as the dimensionality increases, it is difficult for DGEA to find solutions that have converged well to guide the population's evolution. LMOEA-DS has the best performance on LSMOP3, LSMOP6, and LSMOP9. In our experiments, these three problems require MOEAs to have a very high diversity maintenance strategy. The original intention of the pro-

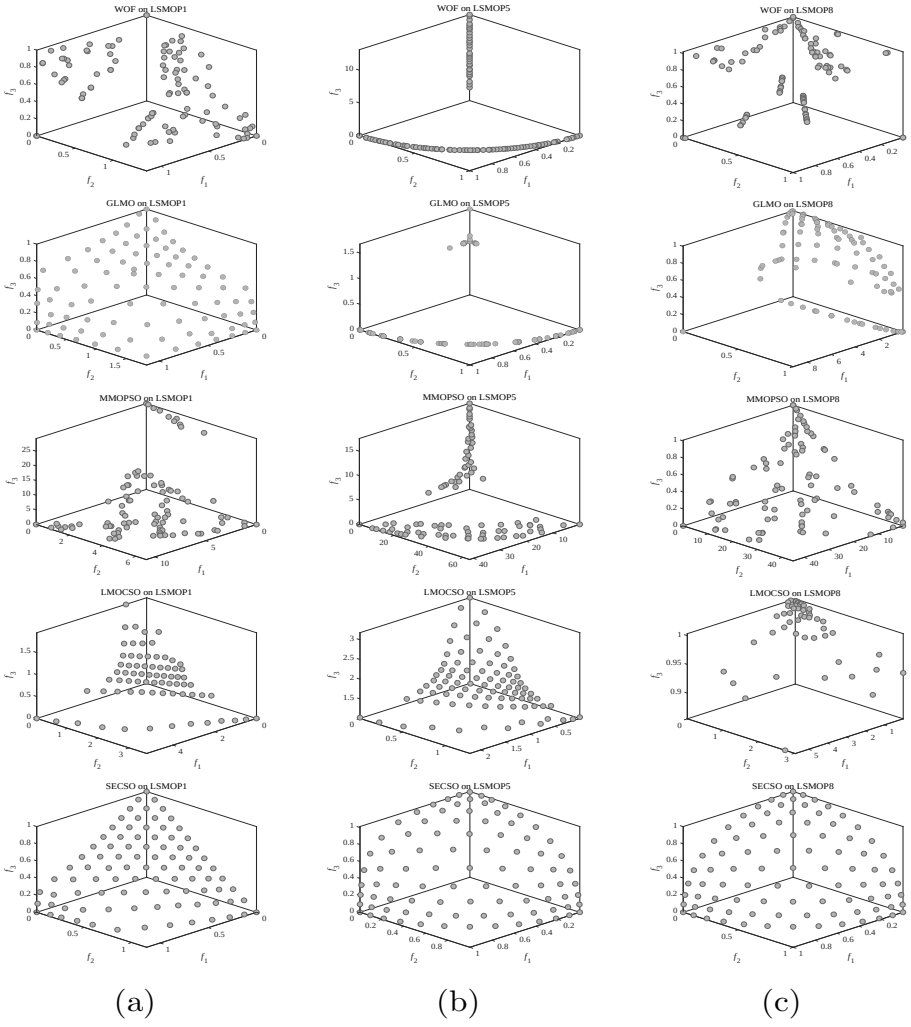

Figure 10: Nondominated solutions with the median IGD values obtained by WOF, MOEA/DVA, MMOPSO, LMOCSO and SECSO on tri-objective LSMOP1, LSMOP5 and LSMOP8 with 5,000 decision variables.

posed SECSO was to improve the convergence speed of the CSO algorithm, so it is relatively inferior to LMOEA-DS on these three problems. Interestingly, MOEA/D-GR does not have a strategy, especially in dealing with large-scale decision variables but has a good performance. MOEA/D-GR is better than the large-scale optimizer DGEA in many test examples and achieved eight optimal values in this experiment. Decomposition-based optimizers may have

great potential for solving large-scale optimization problems.

Table 4: STATISTICS OF IGD RESULTS OBTAINED BY FIVE COMPARED ALGORITHMS ON 36 TEST INSTANCES FROM LMF TEST SUITE. THE BEST RESULT IN EACH ROW IS HIGHLIGHTED..

Problem	M	D	WOF	GLMO	MMOPSO	LMOCSO	SECSO
LMF1	3	100	7.0167e-1 (2.16e-2)−	6.6123e-1 (1.46e-1)−	2.3151e+0 (1.00e-1)−	5.9218e-1 (4.54e-2)−	3.5415e-1 (6.85e-3)
		200	6.7487e-1 (1.28e-2)−	4.6446e+1 (3.23e+1)−	7.0734e-1 (1.54e-2)−	9.6567e-1 (4.01e-2)−	4.7133e-1 (6.60e-2)
		500	6.8046e-1 (7.22e-3)−	9.1068e+0 (1.06e+1)−	3.3139e+1 (3.36e+1)−	7.5792e+0 (3.12e+0)−	4.1363e-1 (9.66e-3)
LMF2	3	100	2.5464e-1 (1.37e-2)−	3.5393e-1 (8.18e-2)−	4.3080e-1 (4.96e-2)−	3.4804e-1 (2.06e-1)−	1.7033e-1 (3.56e-2)
		200	2.5003e-1 (6.42e-3) +	3.8530e-1 (5.30e-2)−	5.8616e-1 (4.97e-2)−	3.2555e-1 (1.63e-1)≈	3.3497e-1 (8.37e-2)
		500	2.5343e-1 (7.16e-3) +	2.9561e-1 (8.13e-2)+	6.9165e-1 (2.68e-2)−	5.4639e-1 (1.95e-1)−	4.2378e-1 (3.36e-2)
LMF3	3	100	5.9733e-1 (2.64e-1)−	1.1506e+0 (2.02e-1)−	7.8238e-1 (9.22e-2)−	5.0899e-1 (1.09e-1)−	4.7269e-1 (2.23e-1)
		200	3.1190e+0 (1.93e-1)−	2.9873e+0 (8.51e-1)−	2.6768e+0 (1.52e+0)−	3.9961e-1 (3.91e-2) +	1.0065e+0 (2.42e-1)
		500	1.2841e+1 (1.29e+1)−	1.4875e+1 (1.46e+1)−	2.1968e+1 (1.42e+1)−	1.8576e+0 (2.09e-1)−	9.2585e-1 (1.92e-1)
LMF4	3	100	5.2997e-1 (9.93e-2)−	5.1039e-1 (3.22e-2)−	1.1726e+0 (1.46e-1)−	6.0752e-1 (8.07e-2)−	4.1052e-1 (1.88e-1)
		200	4.7971e-1 (2.39e-2)−	6.3111e-1 (9.54e-2)−	2.0944e+0 (4.64e-1)−	7.3986e-1 (1.85e-2)−	4.0725e-1 (7.28e-2)
		500	7.1646e-1 (1.81e-2)−	7.5318e-1 (6.34e-2)−	5.2779e+0 (3.18e-1)−	1.6807e+0 (1.70e-1)−	5.3876e-1 (4.23e-2)
LMF5	3	100	2.6381e-1 (1.07e-2)−	2.7446e-1 (2.80e-3)−	4.4847e-1 (1.63e-1)−	2.9937e-1 (1.77e-1)−	1.3038e-1 (2.78e-2)
		200	2.4182e-1 (3.32e-3)−	2.4565e-1 (2.28e-3)−	7.3376e-1 (8.86e-2)−	6.0928e-1 (2.90e-3)−	1.6027e-1 (7.18e-2)
		500	3.1711e-1 (6.77e-2)−	3.3556e-1 (6.39e-2)−	8.0009e-1 (6.95e-2)−	1.1135e+0 (6.54e-2)−	2.2838e-1 (4.51e-2)
LMF6	3	100	2.6350e-1 (3.42e-3) +	4.2867e-1 (1.25e-1)−	6.8458e-1 (1.08e-1)−	4.7600e-1 (2.30e-2)−	3.6847e-1 (1.06e-1)
		200	3.8118e-1 (1.19e-1)−	5.3824e-1 (1.03e-1)−	1.2337e+0 (3.56e-1)−	3.7191e-1 (2.80e-2)−	3.1442e-1 (1.36e-2)
		500	4.3884e-1 (1.90e-2) +	7.9396e-1 (4.11e-1)≈	1.6462e+0 (6.93e-2)−	8.2072e-1 (1.56e-1)−	7.8392e-1 (5.63e-2)
LMF7	3	100	6.9007e-1 (9.14e-2)−	7.7029e-1 (3.92e-1)−	1.2258e+0 (3.34e-1)−	7.8609e-1 (1.46e-1)−	6.1992e-1 (1.81e-2)
		200	6.7021e-1 (1.72e-2)−	7.7855e-1 (8.50e-2)−	2.8615e+0 (4.99e-1)−	9.6062e-1 (3.45e-1)−	6.4957e-1 (9.57e-2)
		500	8.9177e-1 (3.06e-3)−	1.4088e+0 (7.39e-1)−	1.6472e+1 (5.14e+0)−	2.7549e+0 (9.90e-2)−	8.0660e-1 (3.32e-2)
LMF8	3	100	9.2744e-1 (4.34e-1)−	1.2650e+0 (2.63e-1)−	1.2919e+0 (3.97e-1)−	5.4243e-1 (3.57e-2) +	6.8392e-1 (4.48e-2)
		200	6.0630e-1 (1.33e-1) +	9.5165e-1 (9.87e-2)≈	1.5071e+0 (3.23e-1)≈	7.9135e-1 (1.06e-1)+	1.1675e+0 (8.55e-2)
		500	7.3330e-1 (1.90e-1) +	1.4346e+0 (8.94e-1)≈	2.1789e+0 (7.49e-2)≈	1.8355e+0 (4.28e-2)≈	1.6443e+0 (2.93e-1)
LMF9	3	100	4.6965e+0 (5.27e+0)−	1.3488e+0 (1.66e-1)−	1.8453e+1 (9.01e+0)−	7.4516e-1 (2.09e-1) +	9.1483e-1 (6.22e-2)
		200	1.2106e+0 (4.27e-2) ≈	2.2183e+1 (2.97e+1)−	5.4216e+1 (2.38e+1)−	3.2377e+0 (5.54e-1)≈	2.8573e+0 (4.22e-2)
		500	1.0084e+0 (3.81e-1) +	1.0449e+0 (4.23e-2)+	5.2221e+2 (2.48e+2)−	3.1686e+1 (5.28e+0)≈	2.8236e+1 (1.21e+0)
LMF10	3	100	8.6071e-1 (1.24e-3) +	1.4550e+0 (8.39e-1)≈	9.4896e+0 (6.79e+0)−	1.1903e+0 (2.26e-1)≈	1.8822e+0 (4.87e-1)
		200	7.2422e-1 (1.93e-1) +	1.3845e+0 (2.69e-1)+	2.3274e+1 (8.80e+0)−	7.3391e+0 (1.02e+0)−	5.5232e+0 (6.59e+0)
		500	9.1566e+0 (1.20e+1)+	3.3462e+0 (2.57e+0)+	4.3432e+2 (5.52e+1)−	9.1342e+1 (5.21e+0)−	1.9676e+1 (2.35e+1)
LMF11	3	100	6.4130e-1 (7.49e-2)−	7.2161e-1 (2.79e-3)−	1.1076e+0 (2.39e-1)−	4.4138e-1 (5.10e-2)−	3.7440e-1 (1.97e-2)
		200	6.1457e-1 (6.42e-2)−	7.3557e-1 (1.30e-1)−	1.7883e+0 (2.99e-1)−	6.6608e-1 (1.51e-1)−	5.0134e-1 (5.40e-2)
		500	7.5051e-1 (6.16e-2)−	7.9456e-1 (1.54e-2)−	2.6441e+0 (3.07e-1)−	8.7193e-1 (1.12e-1)−	4.8804e-1 (2.36e-2)
LMF12	3	100	9.6936e-1 (3.70e-2) +	1.2167e+0 (4.56e-3)≈	1.9972e+0 (3.64e-2)≈	1.5698e+0 (5.40e-1)≈	1.1519e+0 (6.03e-1)
		200	1.0360e+0 (9.59e-3)≈	1.1734e+0 (3.95e-2)≈	7.6991e+0 (2.06e+0)−	4.6671e+0 (8.65e-1)−	2.4564e+0 (1.97e+0)
		500	1.0652e+0 (7.18e-3) +	1.1508e+0 (5.35e-2)+	7.0382e+1 (4.01e+1)−	2.1703e+1 (5.52e+0)≈	1.2960e+1 (1.53e+0)
+/ − / ≈			12/22/2	5/25/6	0/33/3	4/25/7	

The nondominated solutions with the median IGD values of the five algorithms on LSMOP1, LSMOP5, and LSMOP8 are given in Fig. 9 and Fig. 10 to clearly show the performance advantages of SECSO on the LSMOP problem. The optimization results of MMOPSO are far from the true PF. LMOCSO did not converge due to insufficient function evaluations. WOF, GLMO, MMOPSO, and LMOCSO show inferior distribution. Although WOF and GLMO can find a set of good convergence solutions compared to LMOCSO and MMOPSO for the three test problems, the convergence and uniformity of the solutions obtained by SECSO are significantly better than WOF.

In order to test the optimized performance of SECSO under complex landscapes, we tested nine MOEAs on LMF1-12. We display the mean IGD and HV values obtained in Table 4 and Table 5, respectively. Due to the page limits, we only show the test results of the tri-objective LMF1-12. Compared with the bi-objective LMF problem, x^p in $H(x^p)$ is divided into two groups

of overlapping variables. In addition, $g_1(x^{d,1})$, $g_2(x^{d,2})$, and $g_3(x^{d,3})$ in $f_1(x)$, $f_2(x)$, and $f_3(x)$ are also completely different, making the landscape more complicated. It will make it more difficult for large-scale MOEAs to maintain population diversity and rapidly converge compared to the bi-objective LMF problem.

Table 5: STATISTICS OF HV RESULTS OBTAINED BY FIVE COMPARED ALGORITHMS ON 36 TEST INSTANCES FROM LMF TEST SUITE. THE BEST RESULT IN EACH ROW IS HIGHLIGHTED.

Problem	M	D	RVEA	LMOEA-DS	DGEA	MOEA/D-GR	SECSO
LMF1	3	100	7.8056e-2 (6.13e-2)−	3.4716e-2 (3.80e-2)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	1.4992e-1 (1.05e-1)
		200	0.0000e+0 (0.00e+0)−	3.1554e-2 (7.06e-2)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	1.1225e-1 (1.12e-1)
		500	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	9.1294e-2 (1.05e-1)
LMF2	3	100	5.2994e-1 (3.49e-2)−	7.2328e-1 (7.05e-2)−	0.0000e+0 (0.00e+0)−	5.3842e-1 (1.59e-2)−	8.0439e-1 (6.99e-2)
		200	3.5473e-1 (9.60e-2)−	7.5873e-1 (2.58e-2)≈	0.0000e+0 (0.00e+0)−	4.2926e-1 (9.95e-2)−	7.5965e-1 (6.02e-2)
		500	2.5462e-1 (1.81e-2)−	7.8147e-1 (6.70e-2)≈	0.0000e+0 (0.00e+0)−	2.4428e-1 (1.15e-2)≈	6.3089e-1 (1.37e-1)
LMF3	3	100	1.0019e-1 (5.63e-2)−	2.5123e-2 (5.02e-2)−	0.0000e+0 (0.00e+0)−	2.6434e-1 (1.67e-2)≈	2.2378e-1 (1.85e-1)
		200	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	6.0252e-2 (3.01e-2)+	2.6687e-2 (4.11e-2)
		500	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	2.0212e-2 (3.65e-2)
LMF4	3	100	9.1317e-2 (4.45e-3)−	4.8055e-2 (3.49e-2)−	0.0000e+0 (0.00e+0)−	1.7230e-2 (1.60e-2)−	1.4236e-1 (5.37e-2)
		200	0.0000e+0 (0.00e+0)−	3.5693e-2 (4.11e-2)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	1.3943e-1 (2.70e-2)
		500	0.0000e+0 (0.00e+0)−	2.2706e-3 (4.41e-3)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	4.0859e-2 (3.60e-2)
LMF5	3	100	5.9049e-1 (7.59e-2)−	6.3347e-1 (1.74e-1)−	0.0000e+0 (0.00e+0)−	3.7912e-1 (5.57e-2)−	8.7067e-1 (1.90e-2)
		200	1.9566e-1 (1.25e-1)−	5.2160e-1 (1.95e-1)−	0.0000e+0 (0.00e+0)−	8.0298e-2 (1.55e-2)−	8.7336e-1 (8.08e-2)
		500	0.0000e+0 (0.00e+0)−	6.7815e-1 (1.06e-1)≈	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	5.6951e-1 (3.81e-1)
LMF6	3	100	2.6188e-1 (5.68e-2)≈	2.0615e-1 (6.14e-2)≈	0.0000e+0 (0.00e+0)−	2.1656e-1 (4.58e-2)≈	3.5669e-1 (7.22e-2)
		200	1.6541e-1 (1.03e-1)−	2.5033e-1 (6.85e-2)−	0.0000e+0 (0.00e+0)−	5.8478e-2 (3.97e-2)−	3.0386e-1 (2.06e-1)
		500	3.4711e-3 (3.96e-3)−	1.1181e-1 (9.23e-2)+	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	1.5746e-2 (2.73e-2)
LMF7	3	100	2.5099e-2 (2.74e-2)−	8.3687e-3 (1.67e-2)−	0.0000e+0 (0.00e+0)−	5.7449e-5 (8.12e-5)−	6.9264e-2 (4.82e-2)
		200	8.5736e-3 (1.07e-2)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	2.3887e-2 (1.23e-2)
		500	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	1.3385e-2 (1.82e-2)
LMF8	3	100	7.1175e-2 (4.82e-2)≈	1.1640e-1 (2.85e-2) +	0.0000e+0 (0.00e+0)−	2.4205e-3 (2.84e-3)−	7.7876e-2 (6.02e-2)
		200	2.2740e-3 (3.35e-3)−	4.9792e-2 (7.65e-2) +	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	3.4949e-3 (6.99e-3)
		500	0.0000e+0 (0.00e+0)≈	5.2637e-3 (9.12e-3) +	0.0000e+0 (0.00e+0)≈	0.0000e+0 (0.00e+0)≈	0.0000e+0 (0.00e+0)
LMF9	3	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)−	2.9555e-3 (5.35e-3)
		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)
		500	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)
LMF10	3	100	5.3850e-4 (1.08e-3) +	0.0000e+0 (0.00e+0)≈	0.0000e+0 (0.00e+0)≈	0.0000e+0 (0.00e+0)≈	0.0000e+0 (0.00e+0)
		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)−	3.7101e-3 (4.32e-3)
		500	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)
LMF11	3	100	1.0929e-1 (2.16e-2)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	1.1715e-1 (3.67e-2)
		200	6.0965e-2 (4.65e-2)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	6.2900e-2 (2.44e-2)
		500	1.0799e-2 (1.87e-2)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	0.0000e+0 (0.00e+0)−	7.0372e-2 (1.26e-2)
LMF12	3	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)
		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)
		500	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)
+/ − / ≈			1/26/9	4/21/11	0/28/8	1/24/11	

Table 6: STATISTICS OF HV RESULTS OBTAINED BY FIVE COMPARED ALGORITHMS ON 36 TEST INSTANCES FROM LMF TEST SUITE. THE BEST RESULT IN EACH ROW IS HIGHLIGHTED.

Problem	M	D	DGEA	LMOCSO	MMOPSO	MOEA/D-GR	SECSO
mDTLZ1	3	100	1.3997e+2 (8.07e+1)≈	1.8361e+2 (6.27e+1)−	2.0660e+2 (5.16e+1)−	1.0480e+2 (6.99e+1)≈	7.2188e+1 (1.95e+1)
mDTLZ2			3.0587e-1 (5.03e-2)≈	2.2923e-1 (3.52e-2)+	1.9333e-1 (8.67e-3) +	2.1784e-1 (5.04e-3)+	3.6985e-1 (1.64e-2)
mDTLZ3			9.3093e+1 (4.06e+1)≈	1.3862e+2 (3.47e+1)≈	1.7233e+2 (6.00e+1)−	9.1249e+1 (4.79e+1)≈	7.3112e+1 (3.61e+1)
mDTLZ4			2.0074e-1 (2.37e-3)−	2.0621e+0 (3.91e-1)−	1.2797e+0 (1.55e-1)−	2.4678e+0 (1.07e-1)−	1.6240e-1 (2.69e-2)
+/ − / ≈			0/1/3	1/2/1	1/3/0	1/1/2	

In particular, based on the Wilcoxon rank-sum test, when respectively compared to WOF, GLMO, MMOPSO, LMOCSO, RVEA, LMOEA-DS,

DGEA, and MOEA/D-GR, the proposed SECSO is better in 22, 25, 33, 25, 26, 20, 28, and 24 out of 36 cases and is only worse in 15, 5, 0, 4, 1, 5, 0, and 1 cases. Compared with WOF, the proposed SECSO has better performance on LMF1-7 and LMF12. LMF1-7 uses the same base function to construct $g_1(x^{d,1})$ and $g_3(x^{d,3})$, while in LMF11, $g_1(x^{d,1})$, $g_2(x^{d,2})$ and $g_3(x^{d,3})$, the basic functions used are completely different. We can infer that SECSO is not sensitive to the basic function construction type of problem. This is mainly due to the self-exploratory strategy that improves the convergence speed of CSO and the ability to jump out of the local optimum regions.

Table 7: STATISTICS OF HV RESULTS OBTAINED BY FIVE COMPARED ALGORITHMS ON 36 TEST INSTANCES FROM LMF TEST SUITE. THE BEST RESULT IN EACH ROW IS HIGHLIGHTED.

Problem	M	D	DGEA	LMOCSO	MMOPSO	MOEA/D-GR	SECSO
WOSGZ1	2	100	1.6044e-2 (3.56e-4)−	5.7675e-3 (5.46e-4) +	1.1176e-2 (7.35e-4)−	1.6014e-2 (2.21e-3)−	9.2698e-3 (3.86e-4)
WOSGZ2			1.8083e-2 (7.97e-4)−	7.0289e-3 (8.66e-4) +	1.1668e-2 (5.89e-4)−	1.9545e-2 (7.03e-4)−	9.7723e-3 (1.51e-3)
WOSGZ3			4.2725e-2 (4.80e-3)−	1.5157e-2 (2.40e-3)−	1.2932e-2 (1.04e-3)−	2.5770e-2 (1.28e-3)−	1.1355e-2 (3.18e-3)
WOSGZ4			4.7685e-2 (3.48e-3)−	2.0215e-2 (2.50e-3)≈	1.3471e-2 (5.68e-4) +	3.2521e-2 (1.06e-3)−	2.2481e-2 (4.99e-3)
WOSGZ5			6.9898e-2 (2.40e-2)−	3.0761e-2 (3.99e-3)−	2.0411e-2 (1.92e-3)≈	9.9087e-2 (3.28e-3)−	1.8702e-2 (2.07e-3)
WOSGZ6			8.1614e-2 (2.13e-2)−	4.3057e-2 (1.29e-3)−	2.7759e-2 (3.90e-3)≈	1.0357e-1 (4.03e-3)−	2.7496e-2 (9.83e-3)
WOSGZ7			1.7725e-1 (6.23e-3)−	1.8912e-1 (1.81e-3)−	1.6287e-1 (1.08e-2)≈	1.6597e-1 (4.79e-3)≈	1.6172e-1 (6.01e-3)
WOSGZ8			4.2069e-2 (5.84e-3)≈	2.9941e-2 (1.64e-3)+	1.3430e-2 (2.43e-3) +	5.1567e-2 (3.71e-3)≈	4.9288e-2 (1.24e-3)
WOSGZ9	3	100	4.1734e-2 (9.47e-5)−	4.1165e-2 (4.19e-6)−	8.4182e-2 (4.77e-3)−	5.0976e-2 (1.12e-3)−	4.1086e-2 (1.54e-5)
WOSGZ10			4.1666e-2 (1.29e-4)−	4.1168e-2 (5.05e-6)−	7.6292e-2 (6.63e-3)−	5.3551e-2 (1.35e-3)−	4.1095e-2 (3.00e-5)
WOSGZ11			4.2068e-2 (8.38e-5)−	4.1185e-2 (1.31e-6)−	8.1605e-2 (1.18e-2)−	5.6956e-2 (2.35e-3)−	4.0270e-2 (3.03e-5)
WOSGZ12			4.2184e-2 (9.34e-5)−	4.1182e-2 (5.25e-6)−	8.3132e-2 (9.12e-3)−	5.3230e-2 (1.47e-3)−	4.0329e-2 (5.47e-5)
WOSGZ13			4.6767e-2 (8.22e-4)−	4.1361e-2 (1.33e-5) +	1.0130e-1 (5.66e-3)−	8.2761e-2 (2.97e-3)−	4.2411e-2 (4.01e-4)
WOSGZ14			4.7490e-2 (1.19e-3)−	4.1464e-2 (1.86e-5) +	1.0545e-1 (3.53e-3)−	8.3430e-2 (2.79e-3)−	4.2272e-2 (4.47e-4)
WOSGZ15			8.0206e-2 (1.80e-3)−	8.4027e-2 (2.12e-3)−	1.8928e-1 (1.04e-2)−	2.0648e-1 (1.13e-2)−	5.6895e-2 (1.36e-3)
WOSGZ16			3.6728e-2 (2.67e-4)≈	3.7193e-2 (2.34e-4)−	4.8387e-2 (3.34e-3)≈	7.0191e-2 (4.90e-3)−	3.6633e-2 (1.99e-4)
+/ − / ≈			0/14/2	5/10/1	2/10/4	0/14/2	

Table 6 shows the IGD values of SECSO and DGEA, LMOCSO, MMOPSO, and MOEA/D-GR in optimizing mDTLZ1-mDTLZ4 problems. It can be seen that MMOPSO achieves one optimal value while SECSO achieves three optimal values. As the dimension of decision variables increases, the hard-dominated boundary property of mDTLZ makes it difficult for the five algorithms to converge on both mDTLZ2 and mDTLZ4. However, the proposed SECSO achieves better IGD values than LMOCSO, a CSO variant. Table 7 shows the IGD values of SECSO and DGEA, LMOCSO, MMOPSO, and MOEA/D-GR in optimizing WOSGZ1-WOSGZ16 problems. The CSO-based variant showed potential in addressing the WOSGZ test suite. Out of sixteen questions, the CSO-based variant achieved fourteen optimal values. Among them, LMOCSO achieved four optimal values, and SECSO achieved ten optimal values. It can be seen that the proposed SECSO has advantages

over the CSO-based variant.

5. CONCLUSION

In SECSO, we have proposed a particle self-exploratory strategy, which allows the loser to obtain derived particles through self-exploratory in the subspace. The derivative radius of the particles is adjusted adaptively according to the evolution process of the population and the fitness of the particles. Under the particle self-exploratory mechanism, the particles with better fitness bear the responsibility of exploration. The particles with poorer fitness are used for exploitation. Later, Our strategy combines the mechanism with CSO into a new learning strategy that first self-exploratory and then learn. The self-exploratory mechanism and CSO complement each other. The self-exploratory mechanism makes up for the shortcomings of CSO in exploitation and exploration, thereby increasing the convergence and diversity of the population and preventing the algorithm from appearing prematurely.

To verify that the proposed SECSO has good computational efficiency and optimum performance in solving LMOPs within a more extensive range of function evaluations, SECSO was compared with the most advanced large-scale MOEAs on LSMOP, mDTLZ, WOSGZ, and the LMF problems. These experiments show that SECSO has apparent advantages in computing efficiency and can find better solutions under different function evaluations.

The proposed SECSO has shown good potential in LMOPs. However, the distribution of the algorithm needs to be improved on a few complex problems. For example, SECSO did not perform well on disconnected PF problems. In the future, our work will focus on using the information in the particle competition stage to enhance the efficiency of particle's exploration.

6. Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Grant No. 62176228 and Grant No. 61876164), in part by the Research Foundation of Education Bureau of Hunan Province, China (Grant No. 21A0444), in part by the Natural Science Foundation of Hunan Province, China (Grant No. 2020JJ4590), in part by the General Project of Hunan Education Department (Grant No. 21C0077), and in part by the Science and Technology Plan Project of Hunan Province (Grant No 2018TP1036)

References

[1] A. Rodríguez-Molina, M. G. Villarreal-Cervantes, E. Mezura-Montes, M. Aldape-Pérez, Adaptive controller tuning method based on online multiobjective optimization: A case study of the four-bar mechanism, IEEE Transactions on Cybernetics.

[2] Y. Xue, Y. Tang, X. Xu, J. Liang, F. Neri, Multi-objective feature selection with missing data in classification, IEEE Transactions on Emerging Topics in Computational Intelligence.

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

[4] M. Kim, T. Hiroyasu, M. Miki, S. Watanabe, SPEA2+: Improving the performance of the strength Pareto evolutionary algorithm 2, in: International Conference on Parallel Problem Solving from Nature, Springer, 2004, pp. 742–751.

[5] Y. Xue, H. Zhu, J. Liang, A. Słowik, Adaptive crossover operator based multi-objective binary genetic algorithm for feature selection in classification, Knowledge-Based Systems 227 (2021) 107218.

[6] S. Qin, C. Sun, Y. Jin, Y. Tan, J. Fieldsend, Large-scale evolutionary multi-objective optimization assisted by directed sampling, IEEE Transactions on Evolutionary Computation.

[7] M. A. Potter, K. A. De Jong, A cooperative coevolutionary approach to function optimization, in: International Conference on Parallel Problem Solving from Nature, Springer, 1994, pp. 249–257.

[8] 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 (2) (2015) 275–298.

[9] 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 (2) (2017) 260–275.

[10] 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 (6) (2019) 949–961.

[11] Y. Tian, X. Zheng, X. Zhang, Y. Jin, Efficient large-scale multiobjective optimization based on a competitive swarm optimizer, IEEE Transactions on Cybernetics 50 (8) (2020) 3696–3708.

[12] C. He, R. Cheng, D. Yazdani, Adaptive offspring generation for evolutionary large-scale multiobjective optimization, IEEE Transactions on Systems, Man, and Cybernetics: Systems.

[13] X. Yang, J. Zou, S. Yang, J. Zheng, Y. Liu, A fuzzy decision variables framework for large-scale multiobjective optimization, IEEE Transactions on Evolutionary Computation.

[14] F. Cheng, F. Chu, L. Zhang, A multi-objective evolutionary algorithm based on length reduction for large-scale instance selection, Information Sciences 576 (2021) 105–121.

[15] S. Liu, J. Li, Q. Lin, Y. Tian, K. C. Tan, Learning to accelerate evolutionary search for large-scale multiobjective optimization, IEEE Transactions on Evolutionary Computation.

[16] Q. Yang, W.-N. Chen, J. Da Deng, Y. Li, T. Gu, J. Zhang, A level-based learning swarm optimizer for large-scale optimization, IEEE Transactions on Evolutionary Computation 22 (4) (2017) 578–594.

[17] R. Cheng, Y. Jin, A competitive swarm optimizer for large scale optimization, IEEE Transactions on Cybernetics 45 (2) (2014) 191–204.

[18] R. Cheng, Y. Jin, M. Olhofer, et al., Test problems for large-scale multiobjective and many-objective optimization, IEEE Transactions on Cybernetics 47 (12) (2016) 4108–4121.

[19] S. Liu, Q. Lin, K.-C. Wong, Q. Li, K. C. Tan, Evolutionary large-scale multiobjective optimization: Benchmarks and algorithms, IEEE Transactions on Evolutionary Computation.

[20] I. Giagkiozis, R. C. Purshouse, P. J. Fleming, An overview of population-based algorithms for multi-objective optimisation, International Journal of Systems Science 46 (9) (2015) 1572–1599.

[21] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-international conference on neural networks, Vol. 4, IEEE, 1995, pp. 1942–1948.

[22] C. Yue, B. Qu, J. Liang, A multiobjective particle swarm optimizer using ring topology for solving multimodal multiobjective problems, IEEE Transactions on Evolutionary Computation 22 (5) (2017) 805–817.

[23] J. J. Liang, A. K. Qin, P. N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE transactions on evolutionary computation 10 (3) (2006) 281–295.

[24] P. Mohapatra, K. N. Das, S. Roy, A modified competitive swarm optimizer for large scale optimization problems, Applied Soft Computing 59 (2017) 340–362.

[25] 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.

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

[27] B. Tucker, The flipped classroom, Education next 12 (1) (2012) 82–83.

[28] J. L. Bishop, M. A. Verleger, et al., The flipped classroom: A survey of the research, in: ASEE national conference proceedings, Atlanta, GA, Vol. 30, 2013, pp. 1–18.

[29] M. Li, S. Yang, X. Liu, Shift-based density estimation for Pareto-based algorithms in many-objective optimization, IEEE Transactions on Evolutionary Computation 18 (3) (2013) 348–365.

[30] 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 (1) (2016) 32–46.

[31] B. Li, K. Tang, J. Li, X. Yao, Stochastic ranking algorithm for many-objective optimization based on multiple indicators, IEEE Transactions on Evolutionary Computation 20 (6) (2016) 924–938.

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

[33] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, IEEE Transactions on Evolutionary Computation 20 (5) (2016) 773–791.

[34] H. Zille, Large-scale multi-objective optimisation: New approaches and a classification of the state-of-the-art.

[35] Z. Wang, Q. Zhang, M. Gong, A. Zhou, A replacement strategy for balancing convergence and diversity in moea/d, in: 2014 IEEE congress on evolutionary computation (CEC), IEEE, 2014, pp. 2132–2139.

[36] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: Empirical results, Evolutionary computation 8 (2) (2000) 173–195.

[37] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable multi-objective optimization test problems, in: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600), Vol. 1, IEEE, 2002, pp. 825–830.

[38] H. Li, Q. Zhang, Comparison between nsga-ii and moea/d on a set of multiobjective optimization problems with complicated pareto sets, IEEE Trans. Evol. Comput. 13 (2) (2009) 284–302.

[39] Z. Wang, Y.-S. Ong, H. Ishibuchi, On scalable multiobjective test problems with hardly dominated boundaries, IEEE Transactions on Evolutionary Computation 23 (2) (2018) 217–231.

[40] Z. Wang, Y.-S. Ong, J. Sun, A. Gupta, Q. Zhang, A generator for multiobjective test problems with difficult-to-approximate pareto front boundaries, IEEE Transactions on Evolutionary Computation 23 (4) (2018) 556–571.

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

[42] J. Bader, K. Deb, E. Zitzler, Faster Hypervolume-Based Search Using Monte Carlo Sampling, in: M. Ehrgott, B. Naujoks, T. J. Stewart, J. Wallenius (Eds.), Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems - Proceedings of the 19th International Conference on Multiple Criteria Decision Making, Auckland, New Zealand, January 7-12, 2008, Vol. 634 of Lecture notes in economics and mathematical systems, Springer, 2008, pp. 313–326.

[43] 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), University of Surrey, 2018.

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

[45] K. Deb, Multi-objective optimization using evolutionary algorithms, Vol. 16, John Wiley & Sons, 2001.