Swarm and Evolutionary Computation 75 (2022) 101152

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Offspring regeneration method based on bi-level sampling for large-scale evolutionary multi-objective optimization ☆

Wei Liu, Li Chen *, Xingxing Hao *, Wei Zhou, Xin Cao, Fei Xie

Department of Information Science and Technology, Northwest University, Xi'an 710119, China

ARTICLE INFO

Keywords:
Evolutionary algorithm (EA)
Multi-objective optimization
Large-scale optimization
Directed sampling

ABSTRACT

In evolutionary multi-objective optimization, the size of search space exponentially expands as the number of decision variables increases, which makes the generation of promising candidate solutions more difficult. For this, in this paper, we propose a bi-level offspring generation architecture, together with a deep offspring sampling method. The offspring generation process is divided into two phases. The first phase uses the general genetic operators to generate the offspring, and then in the second phase, the selected non-dominated solutions are utilized by the proposed deep sampling method to produce offspring again. Specifically, the proposed deep sampling method makes use of the selected non-dominated solutions to establish search directions at first, then solutions are sampled on them. It is expected to take advantage of both offspring generation schemes, thereby balancing the diversity and convergence of the population. Existing large-scale evolutionary algorithms can easily be extended to our proposed bi-level architecture. The experimental results demonstrate the significant advantages of the proposed architecture and sampling method, in comparison with several state-of-the-art large-scale multi\many-objective optimization problems in solving LSMOPs with up to 5000 decision variables.

1. Introduction

The concept of decision space scalability of multi-objective evolutionary algorithms (MOEAs) concerning decision variables has been investigated quite intensively in recent years [1]. And large-scale multi-objective problems (LSMOPs), especially those involving hundreds or thousands of decision variables, widely exist in real-world applications. For example, in the field of machine learning, the training of deep neural networks (DNN) often involves determining millions of weights [2]. In network science, tens of thousands of location variables and power allocation variables are involved in the design of telecommunication networks [3]. Consistent with the fact that the volume of search space exponentially expands as the decision variables increases, most meta-heuristic algorithms suffer from the curse of dimensionality [4,5]. To solve this problem, a variety of algorithms have been proposed for LSMOPs. The existing algorithms can be roughly classified into the following four categories.

1. Divide and Conquer

   The main purpose of this category is first to assign the decision variables into several low-dimensional subgroups that are expected to be solvable, and then optimize each of them using a subpopulation, separately. The algorithms that fall into this category can be further classified into two types in terms of the way they group decision variables, namely, the non-variable-analysis grouping and variable-analysis grouping. For the former, the classical grouping strategies mainly include random grouping, ordered grouping, linear grouping and differential grouping [6–13]. For the latter, decision variables are grouped into different types based on their controlling properties, typically including convergence-related variables and diversity-related variables [14–20].
2. Problem Transformation

   [21] first proposed the problem transformation strategy (WOF) for LSMOPs. It decomposes the decision variables into different groups with each group affiliated with a weight variable.

☆ This work was supported in part by the key special project of National Key R&D Program of China (2020YFC1523301), in part by the Shaanxi Key Research and Development Program Project, China (2019ZDLGY10-01), in part by the Shaanxi Key Research and Development Program Project, China (2019ZDLSF07-02), in part by the Xi'an Major Scientific and Technological Achievements Transformation and Industrialization Project, China (20GXSF0005), and in part by the General Project of Education Department of Shaanxi Provincial Government, China under Grant 21JK0926, in part by the Key Research and Development Program of Shaanxi Province, China (No. 2020KW-068), in part by the National Natural Science Foundation of China under Grant 62106199, 62002290, and 62001385, in part by the China Postdoctoral Science Foundation (Grant No. 2021MD703883).

* Corresponding authors.
E-mail addresses: geq_liuwei@163.com (W. Liu), chenli@nwu.edu.cn (L. Chen), ystar1991@126.com (X. Hao).

https://doi.org/10.1016/j.swevo.2022.101152
Received 25 November 2021; Received in revised form 9 August 2022; Accepted 11 August 2022
Available online 24 August 2022

Then, the variables in each group are optimized simultaneously by optimizing the weight variable. [22] presented a different problem transform method (LSMOF), which optimizes the weight variables along different search directions in decision space. [23] utilized a weight optimization framework with random dynamic grouping to solve LSMOPs with 5000 dimensions (WOF-MMOPSO-RDG). [24] adopted the problem reformation method and decomposition-based method to solve LSMOPs in an iterated manner (iLSMOA). The main advantage of this type of algorithm is that the number of weight variables is much smaller than the original variables, thus the search space can be largely reduced.

3. Search Direction-based Approaches
   This idea was originally applied by [22] and further improved by [25]. To guide the search directions of the algorithm toward the Pareto set (PS), a set of well converged and evenly distributed candidate solutions are selected from the population first. Then, different search directions are emitted from the highest and lowest points of the search space to the candidate solutions, which are expected to intersect with the PS. Random sampling along the search direction is then performed at each iteration. They achieved good convergence with less computational cost.
4. Other MOEAs
   Unlike the most large-scale MOEAs that improve scalability by constructing low-dimensional spaces. [26]proposed a competitive swarm optimizer (LMOCSO) for large-scale MOPs, in which they adopted a new particle updating strategy to generate offspring. Afterwards, Ding proposed a multi-stage knowledge-guided algorithm [27] for solving sparse LSMOPs. He et al. proposed an adaptive offspring generation method to reproduce promising offspring solutions [28]. In addition, GMOEA [29] proposed by He et al. and MOEA-CSOD [30] proposed by Liang et al. generate offspring using generative adversarial networks and distributional adversarial networks, respectively, which are trained on the solutions in the current population, to solve LSMOPs.

Though many large-scale optimization algorithms have been proposed so far, the development of large-scale MOEA is still in its infancy and every algorithm has drawbacks. For example, one of the major disadvantages of grouping strategies based on cooperative co-evolution is that the performance of algorithms depends heavily on the grouping mechanisms. In particular, if two or more highly correlated variables are grouped into different groups, the search directions of the algorithm may be misled, thus the performance of the algorithms could degenerate. Moreover, some variable-analysis grouping strategies like MOEA/DVA [14] and LMEA [15] could cause extra computational costs, and their preponderance deteriorates as the number of decision variables increases with limited computing resources. As for the problem transformation-based MOEAs, they may fall into local optima and lack convergence capability because each variable cannot be changed independently. Furthermore, the essence behind grouping and problem transformation strategy is to reduce the dimension of problems by dividing the original search space into smaller sub-spaces, and then carry out optimization in them. However, it is not easy to find a reasonable division of the variables, which often leads to inaccurate search and relatively high number of fitness evaluations [15,17].

By establishing search directions in the search space, both LSMOF [22] and LMOEA-DS [25] achieved good convergence with limited fitness evaluations. However, the search directions in m-objective LSMOPs could hardly intersect with the $(m-1)$-dimensional PS in the d-dimensional decision space since $m \ll d$ [31]. Thus, extra evaluation may also be wasted in sampling for inappropriate search directions.

It can be learned from the above discussions that selection of right search directions and the appropriate offspring generation strategy are instructive for MOEAs in solving LSMOPs. Given this, in this paper, we propose a bi-level offspring generation architecture, together with a deep offspring sampling method for large-scale multi-objective evolutionary optimization. In our proposed architecture, the offspring generation process is divided into two phases. In the first phase, we use common genetic operators to generate the offspring. In the second phase, those offspring selected from the first phase are utilized to conduct the proposed deep sampling method further. The main contributions of this paper are summarized as follows.

1. A bi-level offspring generation architecture, which divides the offspring generation process into two phases, is proposed. The first phase generates offspring, denoted as offspring I, using general genetic operators aiming to speed up the convergence. Then, the second phase generates another group offspring, denoted as offspring II, via a deep sampling method to promote diversity. Finally, offspring I, II, and the parent population are combined for subsequent environment selection. Existing MOEAs can be easily extended to this bi-level architecture. Our experimental results show the advantages of the bi-level architecture MOEAs, compared to their original versions.
2. We designed a non-dominated solution-based deep sampling method, namely, DSNS, for bi-level architecture. In DSNS, non-dominated sorting is first performed on offspring I. Then, the solutions in the first front are selected to establish search directions. Finally, a specified number of solutions are sampled in the search directions. By using the selected non-dominated solutions, the established search directions can track the PS in a dynamic manner.
3. We extend several existing MOEAs to our proposed bi-level offspring generation architecture and examine them on a variety of LSMOPs in comparison with their original versions. Besides, a new large-scale multi-objective algorithm, LMOEA-DSNS, is proposed by extending LMOEA-DS [25] and modifying the environment selection. The performance of LMOEA-DSNS is examined on a variety of 3- to 12-objective large-scale problems with up to 5000 decision variables in comparison with five state-of-the-art algorithms.

The rest of this paper is organized as follows. In Section 2, we briefly introduce the related work on LSMOPs. The motivation for this work is also elaborated in this section. The framework of the proposed bi-level offspring regeneration method is introduced detailedly in Section 3. Section 4 analyzes the parameter settings, and illustrates the effectiveness of the bi-level architecture, DSNS, and the proposed LMOEA-DSNS. Finally, conclusions and future directions are given in Section 5.

2. Background

2.1. Large-scale multi-objective optimization

Without loss of generality, a LSMOP can be formulated as follows:

$$\begin{cases} \min \boldsymbol{F}(\mathbf{x}) = (f_1(x), \ldots, f_m(x)) \\ \mathbf{x} = (x_1, \ldots, x_n) \in \Omega \end{cases} \tag{1}$$

where Ω is the feasible decision space. x consists of n variables and n is greater than 100 [14,15,26]. $\boldsymbol{F}(\mathbf{x})$ consists of m objective functions that need to be minimized simultaneously. Due to the contradiction among the objectives, a set of incomparable solutions called Pareto optimal solutions, representing the trade-off among different objectives, can be obtained. In this work, we concentrate on LSMOPs with 3 to 12 objectives and 500 to 5000 decision variables.

2.2. Theory of the manifold structure

Suppose the objective functions $f_i(x)$, $i = 1, \ldots, m$ are continuously differentiable at $x^* \in \Omega$. If x^* is a (local) Pareto optimal solution, there will be a vector $\alpha = (\alpha_1, a_2, \ldots, a_m)^T (\|\alpha\|_2 = 1)$ satisfying

$$\sum_{i=1}^{m} \alpha_i \nabla f_i(x^*) = 0 \tag{2}$$

The points satisfying Eq. (2) are Karush–Kuhn–Tucker points. Eq. (2) has $n + 1$ equality constraints and $n + m$ variables $x_1, \ldots, x_n, \alpha_1, \ldots \alpha_m$. Thus, under mild conditions, the distribution of the PS to LSMOP is a continuous $(m - 1)$-dimensional manifold [32,33]. Specifically, under mild conditions, the PS is a piece-wise continuous surface for a continuous 3-objective problem and a piece continuous hyper-surface for a continuous many-objective problem ($m > 3$). Test instances in the LSMOP benchmark [34] satisfy this regularity property.

2.3. Motivation

The process of offspring generation in most MOEAs that utilize the general methods like variation and crossover is directionless, such blind search may slow down the convergence speed of MOEAs, especially when solving large-scale multi-objective optimization problems.

For this, several search direction-based approaches like LSMOF [22] and LMOEA-DS [25] have been proposed as reviewed above. Instead of generating offspring indeterminately, they first establish direction vectors in the solution space, aiming at intersecting with the PS. These intersections are obviously optimal solutions. Then, sampling along the search directions is conducted constantly to find them. In addition, sampling on the search directions equals fixing the search directions on several lines in the search space, thus the scale of the search space can be reduced to a certain degree. The experimental results demonstrated that this kind of sampling can considerably accelerate the convergence speed. However, their potential pitfall is that establishing search directions is restricted to the highest and lowest points, which limits the flexibility of sampling solutions.

Thus, we are motivated to propose a bi-level offspring generation architecture. In the proposed architecture, we divide the offspring generation process into two phases, where general genetic operators and a non-dominated solution-based deep sampling method are used to generate offspring in the first and second phases, respectively. Thus, taking advantage of both offspring generation schemes, namely, the first layer promotes convergence and the second layer promotes diversity, to balance the exploration and exploitation during the evolutionary process.

Moreover, to improve the efficiency of the search direction-based sampling approach, this article proposes a new sampling method, i.e., the DSNS. Instead of establishing search directions by the non-dominated solutions and the lower and upper bound points, DSNS only uses the non-dominated solutions. To be specific, we randomly choose two non-dominated solutions to establish the search directions, aiming at promoting the flexibility of sampling.

3. The proposed algorithm

This chapter consists of three parts. The first part mainly describes the general framework of the two-layer offspring regeneration method. The second part describes the details of the DSNS algorithm. In the third part, a new algorithm LMOEA-DSNS is proposed to solve large-scale multi-objective problems by embedding the bi-level offspring regeneration method into LMOEA-DS and modifying the environment selection algorithm.

Algorithm 1: AlG_BI

Input: N (population size)
Output: P' (final population)
1 P_0 = Initialize(N);
2 $t = 0$;
3 **while** *total evaluations is not used up* **do**
4 Q_{t1} = Generate the offspring using P_t ;
5 Perform non-dominated sorting and find the first front Q_f;
6 **if** $|Q_f| = 1$ **then** Q_{t2} = **DSNS**(Q_{t1});
7 **if** $|Q_f| > 1$ **then**
8 Q_{t2} = **DSNS**(Q_f);
9 $Q_t = Q_{t1} \cup Q_{t2}$ // Combine Q_{t1} and Q_{t2} as final offspring;
10 P_{t+1}= **Environmental_Selection**(P_t, Q_t);
11 $t = t + 1$;
12 **end**
13 Output the final population P'

3.1. Main architecture

The details of the bi-level offspring generation method are presented in Algorithm 1. In the first layer, the population Q_{t1} is generated by general operators such as crossover and mutation that are used in the original algorithms (steps 4). Then, non-dominated sorting is performed on Q_{t1} and the solutions in the first front are selected as Q_f (steps 5). The Q_f or Q_{t1} is then used by the proposed sampling method, namely the DSNS that is described in Section B, to generate the second layer offspring population. On the one hand, if there is only one solution in the Q_f, namely, $|Q_f|$ is 1 (steps 6), then all offspring in the first layer will be used in DSNS. This is because the DSNS cannot establish search directions by using one single solution. In this way, the diversity of subsequent sampling solutions can be increased, since solutions in the first layer are randomly chosen to establish the search directions. On the other hand, if $|Q_f|$ is greater than 1 (steps 7), only Q_f is used by DSNS (steps 8). After the DSNS is conducted, we get offspring Q_{t2}. Finally, Q_{t1} and Q_{t2} are combined to carry out the subsequent environmental selection (steps 10). Specifically, many other MOEAs can be extended to this bi-level offspring generation architecture by applying the proposed DSNS to sample solutions further. In the later sections, we embed DSNS into several state-of-the-art MOEAs, namely, NSGA-II [35], RVEA [36], GDE3 [37], LMOCSO, and LMOEA-DS, extending them into bi-level offspring generation architecture. The experimental results show their superior performance compared to their original versions.

3.2. Directed sampling based on non-dominated solutions

Algorithm 2 presents the pseudocode of the DSNS for establishing the search directions. In order to improve the diversity of the population, a total of C_n^2 pairwise combinations of solutions are put into the archive O first. Then, N_c combinations are chosen randomly from it. N_c empirically takes the number of non-dominated solutions in Q_{t1}. Experiments and detailed analysis on choosing the appropriate value for N_c can be found in Table 1 of the Supplementary. Assume O_{t1} and O_{t2} are two non-dominated solutions in the tth selected pair, where $1 \le t \le |O|$, the midpoint $\mathbf{c}_i$ between O_{t1} and O_{t2} is calculated first in line 9 of the Algorithm 2, and identified as the starting point that sends two search vectors to O_{t1} and O_{t2}, respectively. Suppose O_{t1}, O_{t2} are x_p and x_q, respectively, where $p, q \in 1, 2, \ldots, N_c$. The search vectors are promising search directions, which can be defined as follows:

$$\begin{aligned} \boldsymbol{v}_i^+ &= \mathrm{x}_p - \mathbf{c}_i \\ & \qquad for \quad i \in 1, \ldots, N_c \\ \boldsymbol{v}_i^- &= \mathrm{x}_q - \mathbf{c}_i \end{aligned} \tag{3}$$

where $\boldsymbol{v}_i^+$ and $\boldsymbol{v}_i^-$ represent the promising search direction from $\mathbf{c}_i$ to x_p and x_q, respectively. Fig. 1(a) illustrates how the search directions are

Algorithm 2: DSNS

Input: Q_{t1} (the first non-dominated population), σ (the parameter that determines the scope of the search directions), **L** (the lower bound point of the decision space), **U** (the upper bound point of the decision space), N_s (the total number of solutions sampled on the two search directions)

Output: Q_{t2} (the second non-dominated population)

1 n = The number of solutions in Q_{t1};
2 N_c = The number of selected non-dominated pairs, where $N_c = n$ in this paper;
3 O = All pairwise combinations of solutions in Q_{t1};
4 $L_{max} = \|\mathbf{U} - \mathbf{L}\|$;
5 $\mathbf{c} = \emptyset$;
6 $\mathbf{D} = \emptyset$;
7 **for** $i = 1, ..., N_c$ **do**
8 t = Randomly choose a number in $[1, |O|]$;
9 $\mathbf{c}_i = (O_{t1} + O_{t2})/2$ // Calculate the midpoint of the two solutions O_{t1} and O_{t2} in the t-th combinations from O;
10 $\mathbf{D} = \mathbf{D} \cup (O_{t1} - O_{t2})$;
11 $\mathbf{c} = \mathbf{c} \cup \mathbf{c}_i$;
12 **end**
13 $\mathbf{D}' =$ **Normalize D**;
14 $PopX = \emptyset$;
15 λ = Generated N_s random numbers between -1 and 1;
16 $L = L_{max} * \sigma * \lambda$;
17 **for** $i = 1, ..., N_c$ **do**
18 $\mathbf{S} = L_i \cdot \mathbf{D}' + \mathbf{c}_i$ // Sample N_s solutions according to Eq. (4);
19 $\mathbf{S}'$ = Repair **S** using Eq. (5);
20 $PopX = PopX \cup \mathbf{S}'$;
21 **end**
22 Q_{t2} = **Ndsort**$(PopX)$ // Non-dominated sorting on $PopX$;
23 Output Q_{t2}

generated in 3-dimensional decision space. The blue surface in Fig. 1(a) represents the true PS and the black solid triangle is the midpoint between x_p and x_q. The red solid polygon is the intersection of $\boldsymbol{v}_i^+$ and the true PS. Note that $\boldsymbol{v}_i^+$ and $\boldsymbol{v}_i^-$ are on the same line but in the opposite directions. Therefore, the solution s_i^t, $t \in 1, 2, \ldots, N_s$ sampled on both two search directions can be calculated using the following formula.

$$s_i^t = \mathbf{c}_i + \lambda \cdot \frac{x_p - x_q}{\|x_p - x_q\|} \cdot L_{\max} \cdot \sigma \qquad t \in 1, 2, \ldots, N_s \tag{4}$$

where λ is a random number between -1 and 1, that determines the sampling direction and scope. Note that the order of x_p and x_q in (4) does not affect the sampling results. The $L_{\max} = \|\mathbf{U} - \mathbf{L}\|$ is the maximum length of the decision space. As shown in step 18 of the algorithm 2, L is multiplied by the unit vector D' and then added to $\mathbf{c}_i$ to get a series of vectors starting at $\mathbf{c}_i$, where L is the positive or negative random numbers generated in step 16. So the coordinate of each solution can be determined. For example, in Fig. 1(c), $\boldsymbol{v}_i^+$ and $\boldsymbol{v}_i^-$ are two promising search directions defined by solution x_p and x_q in 2-dimensional decision space. s_i^1, s_i^2, s_i^3 and s_i^4 are four solutions sampled in the search directions.

Due to the uncertainty of the search directions, infeasible solutions, i.e., solutions that extend the lower or upper bounds, may be generated during the sampling process. To reduce their number in advance, parameter $\sigma \in (0, 1]$ is introduced to dynamically adjust the search range along the search directions. As shown in Fig. 1(b), if the primary L_{max} is used, a large invalid search area, namely, line segment $\overline{FD}$ and $\overline{GE}$ will be produced. As a result, many infeasible solutions will be generated if sampling in those areas. If an appropriate parameter σ is chosen to fine-tune the L_{max}, the invalid search area can be reduced to $\overline{FA}$ and $\overline{GB}$ as shown in Fig. 1(b), thus saving unnecessary computational cost.

In addition, λ is used to determine in which direction to sample, i.e., $\overline{\mathbf{c}_i A}$ or $\overline{\mathbf{c}_i B}$, and to further adjust the search range.

As for the infeasible solutions, such as s_i^2 in Fig. 1(c), Eq. (5) is used to repair them. In Eq. (5), s represents an infeasible solution. s^d, $d = 1, 2, \ldots, n$ is the dth decision variable of O, d_{max} and d_{min} are the upper and lower bounds of the dth decision variable, respectively. After repair, the infeasible solution s will be pulled back into the feasible region.

$$s^{d'} = \max\{\min\{s^d, d_{\max}\}, d_{\min}\} \tag{5}$$

After sampling, we end up with $N_c \times N_s$ promising solutions and the same number of fitness evaluations are used, as long as the selected non-dominated solutions are well converged. Then, non-dominated sorting is done on them to produce the non-dominated solutions set Q_{t2}. Finally, Q_{t1} and Q_{t2} are combined as guiding solutions Q_t for the following environment selection.

Note that the proposed sampling method DSNS is different from that used in DGEA. In DGEA, the search directions are established from the start point to the end point, which limits the search scope to around these two points. However, DSNS establishes the search directions at the midpoint of the two random solutions. In addition, the search scope spans through the entire decision space, so the sampled solutions are in principle more diverse.

3.3. Modified LMOEA-DS: LMOEA-DSNS

Algorithm 3: Environment_Selection

Input: Q_t (the guiding solutions), **W** (a set of reference vectors), P_t (the current parent population), N_ε (the threshold that determine which environment selection method will be used)

Output: P_{t+1} (next generation)

1 $R_t = Q_t \cup P_t$;
2 R_t' = Perform reproduction operator on R_t;
3 $U_t = R_t \cup R_t'$;
4 Normalize the objective values of the combined population U_t;
5 Assign each individual in U_t to its closest reference vectors in **W** and select the solution on each cluster with the best performance;
6 **if** *the number of the selected solutions is not less than* N_ε **then**
7 P_{t+1} = the selected solutions;
8 **else**
9 P_{t+1} = the solutions selected by NSGA-II-based environment selection on U_t;
10 **end**
11 Output P_{t+1}

The latest large-scale multi-objective algorithm, LMOEA-DS, uses a similar method to LSMOF to establish search directions with fixed lower and upper bounds and randomly pick points on them to produce offspring with good convergence. Therefore, embedding the two-layer search algorithm on it can further improve the performance of the original algorithm to deal with large-scale multi-objective problems. In addition, we slightly modify the environment selection method and propose a new large-scale multi-objective algorithm: LMOEA-DSNS.

After the guiding solutions Q_t are obtained by the bi-level offspring regeneration method, a modified complementary environment selection is used to maintain the balance between convergence and diversity of the population. The steps of the environment selection method are given in Algorithm 3. It mainly includes two parts, i.e., the reproduction and selection. In the reproduction part, the crossover and mutation operators are performed on the combined population $R_t = P_t \cup Q_t$, getting the mutant population R_t'. Then, in the selection part, the complementary environment strategy from [25] is adopted, which includes

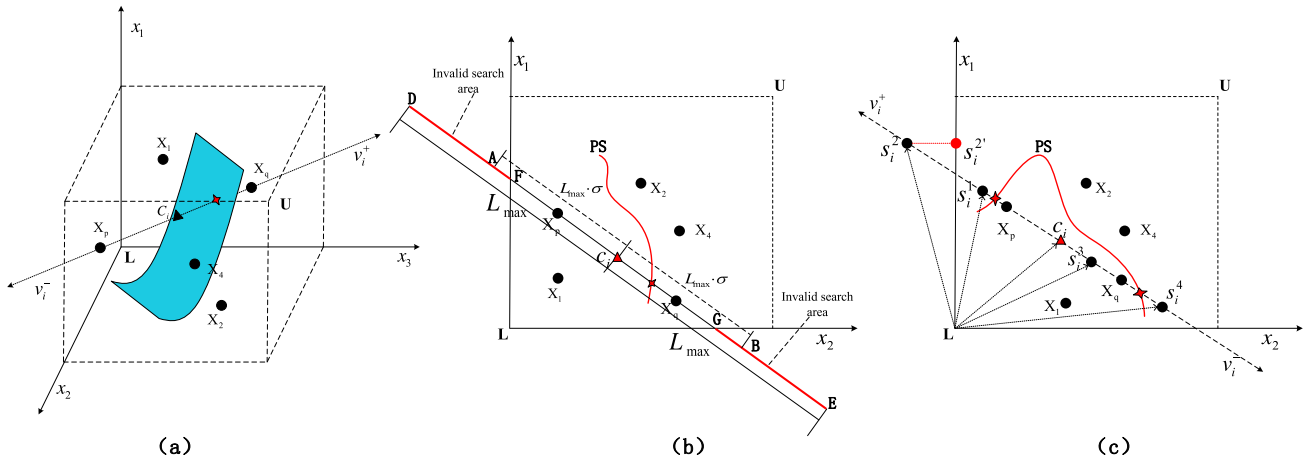

Fig. 1. In (a), two search directions $\boldsymbol{v}_i^+$ and $\boldsymbol{v}_i^-$ emit from midpoint $\mathbf{c}_i$, which are determined by the selected solutions x_p and x_q. (b) depicts of the search scope of the search directions, where $\overline{FG}$ is the valid search area and the red line $\overline{FD}$ and $\overline{GE}$ are invalid search area outside the decision space. (c) depicts the sampling solutions on the two search directions in a 2-D decision space, where **L** and **U** are the upper and lower boundary points, red solid polygons are the intersections between search directions and the PS, and $\mathbf{c}_i$ is the midpoint determined by selected solutions x_p and x_q, s_i^1, s_i^2, s_i^3 and s_i^4 are generated solutions, and $s_i^{2'}$ is the solution mapped from s_i^2 to the decision space.

Table 1
Algorithm parameter settings for **DGEA, LCSA, LMOCSO, WOF, LMOEA-DS** and **LMOEA-DSNS**

DGEA	
Operation of the environmental selection	RVEA
Number of reference vectors for offspring generation	10
LCSA	
The optimization method	NSGA-II
Crossover probability	0.9
Mutation probability	$1/D$[a]
LMOCSO	
Penalty parameter α in APD	2
Crossover probability	1.0
Mutation probability	$1/D$[a]
WOF	
Number of evaluations for original problem	800
Number of evaluations for transformed problem	400
Internal optimization algorithm	NSGA-II
Crossover probability	0.9
Mutation probability	$1/D$[a]
LMOEA-DS	
Cluster number	10
The number of random sampling along each guiding direction	15
Threshold of environment selection N_ϵ	$2/3*N$[a]
Crossover probability	0.9
Mutation probability	$1/D$[a]
LMOEA-DSNS	
Cluster number	10
The number of random sampling along each guiding direction	15
Threshold of environment selection N_ϵ	$2/3*N$[a]
Search directions range parameter σ	0.4
Crossover probability	0.9
Mutation probability	$1/D$[a]

[a] D is the number of decision variables, N is the population size.

two kinds of environment selections, namely, the decomposition-based and NSGA-II-based environmental selections [35], to select the next generation. The difference between the proposed method and that used in [25] is that there is no crossover between the parent population and the guiding solutions, instead, we combine them and perform the crossover and mutation on the combined population subsequently. In addition, the reproduction and selection procedures are only conducted once in this work.

3.4. Advantages

The merits of DSNS are threefold. At first, compared to other solutions, the non-dominated solutions in the current population are closest to the PS. Therefore, the search directions, i.e., the lines in the search space, established by any pair of the non-dominated solutions will also be close to the PS, and solutions sampled on these search directions will be close to the PS as well, even if they do not intersect with the PS. Consequently, unnecessary computational costs can be saved and a good convergence can be guaranteed, since the sampled solutions are all close to the PS. Secondly, DSNS can guarantee a good diversity of the sampled solutions. Imagine if the selected non-dominated solutions are not much different between two generations, the search directions established in [25] will be much the same. This will cause the sampled solutions to be similar to the previous generations, thereby worsening the diversity of the population. But in DSNS, the search directions are determined by randomly paired non-dominated solutions. Therefore, they can be hardly identical to previous generation even if the selected non-dominated solutions are barely the same, thereby avoiding diversity stagnating. Thirdly, the distribution of the PS of a problem in the search space is usually unknown, and PS of different problems may differ greatly. By using the non-dominated solutions to establish the search directions, DSNS can dynamically track the PS of different problems, thus making it more flexible and general.

4. Experimental studies

In this part, we conduct a series of experiments to investigate the proposed bi-level offspring regeneration framework. The inverted generational distance (IGD) [38] and the hyper-volume (HV) are used to evaluate the performance, where in IGD, 10 000 uniformly distributed reference points are selected in the true Pareto front(PF) of each test problem. In HV, the reference point is specified as (1.1, 1.1, …, 1.1) in 3 and 5 objectives in the normalized objective space with the ideal point (0, 0, …, 0) and the nadir point (1.0, …, 1.0), that is, the reference point is 10% larger than the nadir point in every dimension [39]. The smaller the IGD value, the better the performance of the algorithm [40], while the HV indicator is the opposite.

A maximum number of evaluations 80 000 is adopted as the termination criterion for all the compared MOEAs. The population size N is set at 153 for all LSMOPs. Each experiment is run 20 times independently to obtain the statistical results. The Wilcoxon rank sum test [41] with a significance level of 0.05 is adopted to assess the significance of the results obtained by two different algorithms, where

Table 2
The statistical results (mean and standard values of IGD metric) obtained by NSGA-II, RVEA, GDE3, LSMOCSO, LMOEA-DS and their variant on 3-objective LSMOP1-9 with 1000 dimensions. The best result in each row is highlighted in gray.

Problem	M	D	NSGA-II	NSGA-II_BI	RVEA	RVEA_BI	GDE3	GDE3_BI	LMOCSO	LMOCSO_BI	LMOEA-DS	LMOEA-DS_BI
LSMOP1	3	1000	6.0965e+0 (6.67e−1) −	1.6267e+0 (3.49e−1)	4.8145e+0 (3.77e−1) −	1.6517e+0 (1.95e−1)	2.6856e+0 (3.59e−1) −	6.3007e−1 (9.48e−2)	1.4616e+0 (1.01e−1) −	6.3361e−1 (7.60e−2)	4.6657e−1 (3.44e−2) −	4.1255e−1 (3.89e−2)
LSMOP2	3	1000	5.7495e−2 (1.83e−3) +	1.1333e−1 (2.71e−2)	4.3465e−2 (7.98e−5) −	4.2727e−2 (2.26e−4)	5.2972e−2 (6.81e−4) −	5.1361e−2 (9.52e−4)	4.0784e−2 (2.91e−4) −	3.9930e−2 (8.75e−4)	3.8212e−2 (1.15e−3) ≈	3.7596e−2 (1.28e−3)
LSMOP3	3	1000	1.7657e+1 (2.91e+0) −	1.1237e+1 (4.01e−1)	1.1406e+1 (1.90e+0) ≈	1.1124e+1 (5.35e−1)	1.1320e+1 (6.23e−1) −	1.2644e+0 (3.41e−1)	1.3718e+1 (3.11e+0) −	2.5692e+0 (3.16e+0)	8.6192e−1 (2.81e−3) −	8.5525e−1 (1.68e−2)
LSMOP4	3	1000	1.2147e−1 (2.88e−3) +	1.4108e−1 (1.16e−2)	9.9521e−2 (9.21e−4) −	9.6136e−2 (8.51e−4)	1.2665e−1 (1.25e−3) −	9.8848e−2 (3.63e−3)	9.2827e−2 (6.82e−4) −	7.2125e−2 (1.95e−3)	6.8115e−2 (1.41e−3) ≈	6.8658e−2 (2.09e−3)
LSMOP5	3	1000	1.3003e+1 (7.49e−1) −	5.0236e+0 (8.27e−1)	7.8325e+0 (5.19e+0) ≈	3.2607e+0 (3.68e−1)	6.4024e+0 (5.63e−1) −	6.8309e−1 (1.89e−1)	3.1333e+0 (1.11e−1) −	6.4738e−1 (1.93e−1)	5.3235e−1 (1.03e−2) −	5.1570e−1 (5.06e−3)
LSMOP6	3	1000	6.1889e+3 (1.33e+3) −	9.0055e+2 (2.51e+2)	2.0620e+3 (3.52e+2) −	8.0966e+2 (2.28e+2)	2.5171e+3 (1.12e+3) −	1.3374e+0 (3.05e−1)	4.2408e+2 (1.47e+2) −	5.0752e+0 (1.62e+1)	8.1333e−1 (4.35e−2) −	7.5116e−1 (4.86e−2)
LSMOP7	3	1000	1.1081e+0 (2.86e−3) −	1.0458e+0 (1.75e−2)	8.6805e−1 (7.64e−2) +	9.7730e−1 (3.95e−2)	1.0846e+0 (2.61e−3) −	9.0318e−1 (4.48e−2)	1.0551e+0 (2.10e−2) −	9.0828e−1 (5.93e−2)	8.5880e−1 (2.03e−3) −	8.5271e−1 (1.04e−3)
LSMOP8	3	1000	9.3618e−1 (5.57e−2) −	9.2023e−1 (5.44e−2)	8.2518e−1 (1.21e−1) −	5.4249e−1 (4.35e−3)	9.5151e−1 (2.51e−2) −	7.8873e−1 (1.76e−1)	5.9772e−1 (8.80e−2) −	2.5180e−1 (6.39e−2)	2.4226e−1 (4.60e−2) −	1.1765e−1 (3.21e−2)
LSMOP9	3	1000	1.2520e+1 (1.45e+0) +	2.2606e+1 (1.88e+0)	4.9603e+1 (7.33e+0) −	2.7616e+1 (3.26e+0)	2.8379e+1 (1.57e+0) +	3.4761e+1 (2.90e+0)	4.4560e+1 (1.83e+1) ≈	7.2497e+1 (7.39e+1)	5.4103e−1 (3.07e−3) +	5.4410e−1 (1.58e−3)
+/ − / ≈			3/6/0		1/6/2		1/8/0		0/8/1		1/6/2	

the symbols (+), (−), and (≈) indicate that the results obtained by the compared algorithms are significantly better than, worse than, and similar to the proposed algorithm, respectively.

In the rest of this section, the experiment is divided into three parts for detailed comparison. In the first part, we embed the proposed offspring generation method in three representative MOEAs, namely, NSGA-II, RVEA, and GDE3, and two new large-scale multi-objective algorithms LMOCSO and LMOEA-DS, and compared them with their original versions on nine LSMOPs. Then, the effectiveness of the complementary environment selection on the modified algorithm, LMOEA-DSNS, is analyzed in the second part. Finally, LMOEA-DSNS is compared with 5 state-of-the-art large-scale multi-objective optimization algorithms, namely, WOF-NSGA-II [21], LMOCSO [26], DGEA [28], LCSA [42], and LMOEA-DS, which are tailored for large-scale MOPs. Note that because MOEA/DVA [14] and LMEA [15] perform poorly with a small number of fitness evaluations, the comparative results are not presented in this paper. The source codes of the above algorithms are available in PlatEMO [43].

Specific parameter settings of compared large-scale algorithms are summarized in Table 1. They remain in the same settings suggested by their authors. Specifically, to meet the termination criterion, the number of evaluations for the optimization of the original problem and each of the transformed problems in WOF-NSGA-II are set to 800 and 400, respectively. For DGEA [28], we embed it into the RVEA to exhibit its best performance.

4.1. General performance

In the proposed DSNS, the parameter σ in Eq. (4) determines the search range along the established search directions. To set a reasonable value for σ, the effects of $\sigma = [0.2, 1]$ with increasing step 0.1 on 3- and 5-objective LSMOP2, 5 and 8 with up to 5000 decision variables are tested. The test results are depicted in Fig. 2. It can be observed that different settings of σ do not significantly affect the performance of LMOEA-DSNS on 3- and 5-objective LSMOP2. But on 3- and 5-objective LSMOP5 and 8, it has the opposite effects. On the 3-objective LSMOP5 and 8, the performance of LMOEA-DSNS gets worse as the value of σ increases, on the contrary, it gets better on 5-objective LSMOP5 and 8. Specifically, IGD values do not change significantly on 5-objective LSMOP5 and 8 when $\sigma = [0.4, 1]$. As a compromise, we set σ to 0.4 in the following experiments.

To investigate the effectiveness of the proposed bi-level offspring architecture, we embed DSNS into three representative algorithms, i.e., NSGA-II [35], RVEA [36], GDE3 [37], and two latest MOEAs, LMOCSO and LMOEA-DS, which are tailored for large-scale multi-objective problems. Pairwise comparisons between the bi-level versions and their original versions are conducted. The experimental results are presented in Table 2, where Alg_BI denotes the bi-level version of the algorithm Alg, all the compared algorithms are examined on 3-objective LSMOP1 to LSMOP9 with 1000 decision variables. As can be seen from Table 2, Alg_BI algorithms perform better than their original algorithms in most test instances. In particular, NSGA-II_BI and RVEA_BI perform better than their original algorithms in 6 out of 9 instances. The GDE3_BI outperforms its original algorithm in LSMOP1-8. And for large-scale multi-objective algorithms, the LMOCSO_BI and LMOEA-DS_BI greatly improve the performance, they outperform their original version, namely, LMOCSO and LMOEA-DS in 8 and 6 out of the 9 instances, respectively. To sum up, the proposed bi-level offspring generation architecture can improve the performance of the existing MOEAs in solving LSMOPs.

4.2. The effectiveness of the modified complementary environment selection

To verify the effectiveness of the modified complementary environment selection, we compared LMOEA-DSNS with a variant of the LMOEA-DSNS, termed as LMOEA-DSNS-double, which adopts the double reproduction strategy as in LMOEA-DS. Table 3 lists the statistical results obtained by LMOEA-DSNS and LMOEA-DSNS-double on LSMOP2, LSMOP4, and LSMOP6 with up to 5000 variables. It can be seen that LMOEA-DSNS performed significantly better and worse than LMOEA-DSNS-double in 9 and 2 out of 12 instances, respectively. In addition, the IGD results obtained by LMOEA-DSNS and LMOEA-DSNS-double on all 3-, 5-objective LSMOP1-9 test problems are given in Tables 2 and 3, respectively, in the Supplementary, which further prove that the LMOEA-DSNS has better performance than LMOEA-DSNS-double. This indicates that the modified complementary environment selection is effective in the proposed framework. Furthermore, the comparative results of the LMOEA-DS [25] and LMOEA-DS-one that uses our environment selection strategy are provided in Table 4 of the Supplementary, from which we can see that the LMOEA-DS-double also outperforms LMOEA-DS. This further confirms the effectiveness of the proposed modified complementary environment selection strategy.

4.3. Sensitivity analysis of IGD and HV reference point settings

Since the reference point settings for IGD and HV affect the fairness of algorithm comparisons [44], the sensitivity of the IGD and HV reference point settings is analyzed as follows.

The experimental results using 100 000 and 1 000 000 reference points on 12-objective LSMOP1-9 problems with 500 and 1000 decision variables are provided in Tables 11 and 12, respectively, in the Supplementary. It is observed that the final average IGD values obtained by LMOEA-DSNS are still better than those obtained by LCSA, LMOCSO, WOF-NSGA-II, and LMOEA-DS. Compared to DGEA-RVEA, the LMOEA-DSNS underperforms on LSMOP6 and LSMOP8. However, it still has competitive performance on LSMOP1-5 and LSMOP9 test problems. In general, there is little difference between the conclusions we can draw from Tables 11 and 12 and those from Table 10 in the Supplementary, the latter uses 10 000 reference points. In addition, to be consistent with other compared algorithms, we also use 10 000 reference points to calculate IGD values in our experiments.

The statistical results of the HV values obtained by the compared algorithms with reference points r = (1.5, 1.5, ..., 1.5) and r = (2.0, 2.0, ..., 2.0) on 5-, 7-, 9-, 12-objective LSMOP1-9 problems with 500 and 2000 decision variables are given in Tables 13 and 14, and Tables 15 and 16, respectively, in the Supplementary. From comparisons between Tables 13 and 14 (or Tables 15 and 16), we can see that different specifications of reference point, i.e. r= (1.5, ...) and r=(2.0, ...), have little effect on the comparative results among compared algorithms. Furthermore, results in Tables 13–16 in the Supplementary show that the proposed LMOEA-DSNS is still competitive with DGEA-RVEA and LCSA, and outperforms the other three compared algorithms. Interestingly, we found that the performance of LMOEA-DSNS becomes slightly better on 2000-D problems compared to that of 500-D under both reference point settings, but DGEA-RVEA shows the opposite. This demonstrates that LMOEA-DSNS is more effective in high-dimensional problems.

In addition, The statistical results of the HV values obtained by the compared algorithms with reference point r = (1.1, 1.1, ..., 1.1) on 3- and 5-objective LSMOP1-9 problems with up to 5000 decision variables are provided in Tables 6 and 7, respectively, in the Supplementary. As observed from the results, LMOEA-DSNS shows significant advantages over five competitors.

In conclusion, the performance of the proposed LMOEA-DSNS is relatively robust under different IGD and HV reference point settings. It eventually can be verified to be superior or competitive.

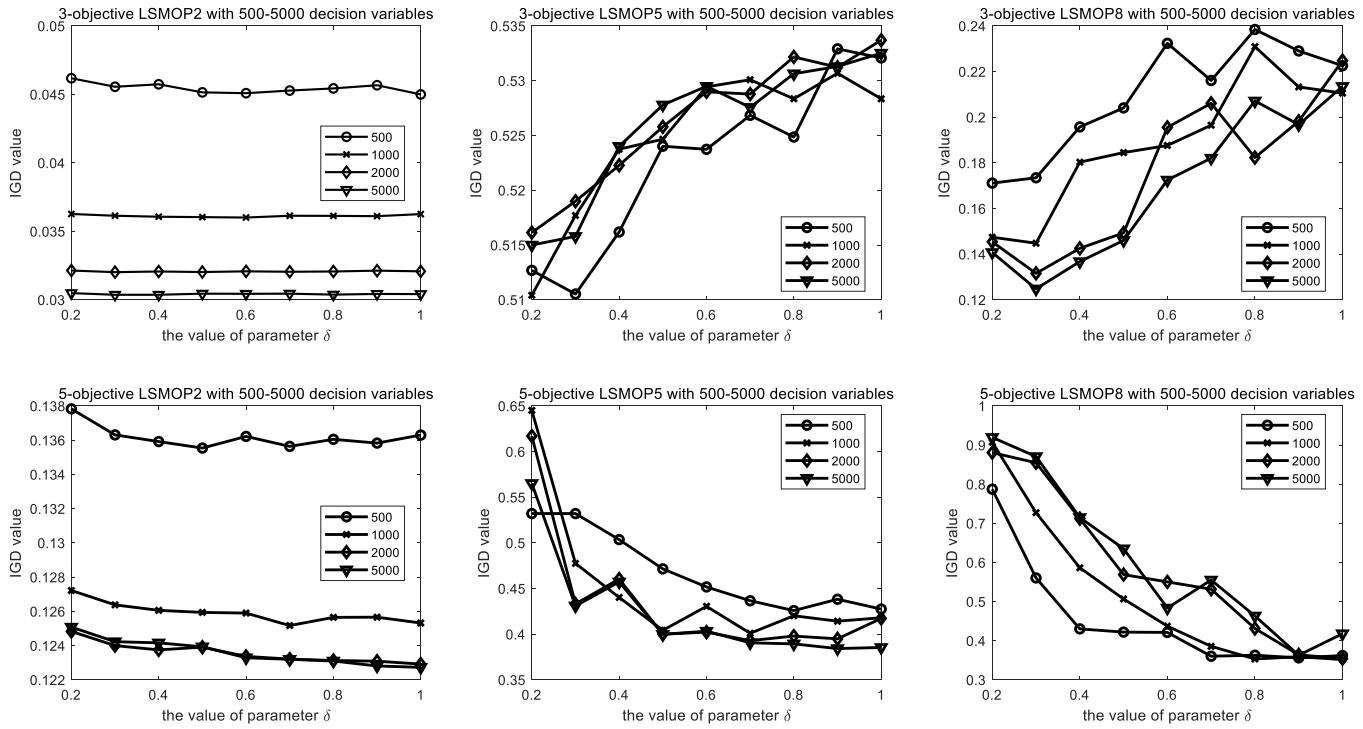

Fig. 2. IGD metric value of LMOEA-DSNS with the value of σ from [0.2, 1] on 3- and 5-objective LSMOP2, LSMOP5 and LSMOP8 with 500 to 5000 decision variables.

Table 3
The statistical results (mean and standard values of IGD metric) obtained by **LMOEA-DSNS-double** and **LMOEA-DSNS** on 3-objective LSMOP2, LSMOP4 and LSMOP6 with 500, 1000, 2000 and 5000 dimensions. The best result in each row is highlighted in gray.

Problem	M	D	LMOEA-DSNS-double	LMOEA-DSNS
LSMOP2	3	500	4.5289e−2 (1.87e−3) ≈	4.5357e−2 (1.41e−3)
	3	1000	3.7271e−2 (1.29e−3) −	3.6062e−2 (3.01e−4)
	3	2000	3.3774e−2 (1.15e−3) −	3.2056e−2 (1.50e−4)
	3	5000	3.1846e−2 (1.11e−3) −	3.0378e−2 (9.83e−5)
LSMOP4	3	500	1.0699e−1 (3.54e−3) +	1.1126e−1 (4.81e−3)
	3	1000	6.8635e−2 (1.61e−3) +	7.0877e−2 (2.17e−3)
	3	2000	4.7742e−2 (1.26e−3) −	4.6943e−2 (1.06e−3)
	3	5000	3.6265e−2 (1.31e−3) −	3.4437e−2 (3.63e−4)
LSMOP6	3	500	8.4783e−1 (1.91e−1) −	7.3402e−1 (2.48e−2)
	3	1000	7.7973e−1 (3.94e−2) −	7.5369e−1 (6.42e−2)
	3	2000	7.8166e−1 (3.34e−2) −	7.4151e−1 (3.11e−2)
	3	5000	8.3261e−1 (1.47e−1) −	7.3022e−1 (2.20e−2)
+/ − / ≈			2/9/1	

4.4. Comparisons with state-of-the-arts

To evaluate the overall performance of the LMOEA-DSNS, we compared LMOEA-DSNS with the aforementioned five state-of-the-art large-scale multi-objective optimization algorithms on 3-, 5-, 7-, 9-, and 12-objective LSMOP1-9 with up to 5000 variables. The IGD results on 3- and 5-objective LSMOPs are presented in Tables 4 and 5, respectively. The remaining comparisons on 7-, 9-, and 12-objective LSMOPs can be found in Tables 8, 9, and 10 of the Supplementary.

Because LMOEA-DSNS is derived from LMOEA-DS [25], the comparative results can be analyzed from two perspectives, i.e., the comparison with LMOEA-DS and the comparisons with other large-scale multi-objective algorithms. As shown in Tables 4 and 5, compared to LMOEA-DS, LMOEA-DSNS obtains significantly better results on 21/36 and 28/36 3-objective and 5-objective instances, respectively, in terms of the IGD values. LMOEA-DS outperforms LMOEA-DSNS only on 3-objective LSMOP4 with 500 and 1000 decision variables, 3-objective LSMOP7 with 5000 decision variables, 3-objective LSMOP9 test instances, and two 5-objective instances. In addition, it can be seen from Tables 8, 9, and 10 of the Supplementary that LMOEA-DSNS obtained 34/36, 25/36 and 36/36 better results in 7-, 9-, and 12-objective problems, respectively, than LMOEA-DS. The above results demonstrate the superiority of LMOEA-DSNS compared to LMOEA-DS.

Compared to other algorithms, LMOEA-DSNS also has obvious advantages. It outperforms WOF in all 3-objective instances and 33/36 5-objective instances. Compared to two recently proposed large-scale many-objective algorithms DGEA and LCSA, LMOEA-DSNS outperforms them on 28/36 and 36/36 3-objective instances, respectively. DGEA-RVEA outperforms LMOEA-DSNS only on LSMOP8 with 500 variables. Regarding the 5-objective LSMOPs, LMOEA-DSNS outperforms DGEA and LCSA on 25/36 and 26/36 instances, respectively. In particular, LMOCSO performs worse than LMOEA-DSNS on all 3- and 5-objective LSMOPs. From these results, we can see that the proposed LMOEA-DSNS can also obtain better results than WOF, DGEA, LCSA, and LMOCSO.

Additionally, LMOCSO fails on all 3- and 5-objective instances compared to LMOEA-DSNS. They represent two different mechanisms for solving large-scale problems. LMOCSO carries out local comparisons between particles. Therefore, the search is circuitous, which can slow down the convergence. However, LMOEA-DSNS, which is equipped with the DSNS, establishes search directions that are likely to intersect with the PS. By randomly sampling solutions on these search directions, LMOEA-DSNS can quickly lock the point nearest to the PS in the whole decision space. Therefore, it can achieve faster convergence than LMOCSO.

The boxplots of the IGD values obtained by DGEA-RVEA, LCSA, LMOCSO, WOF-NSGA-II, LMOEA-DS, and the proposed LMOEA-DSNS on 12-objective LSMOP1-9 test problems with 5000 dimensions are presented in Fig. 3. For clearer visualization, all the original IGD values are processed with logarithmic transformation. We can see that LMOEA-DSNS can obtain relatively low and stable IGD values on most LSMOPs and obtain the best median results on 12-objective LSMOP1,

Table 4
The statistical results (mean and standard values of IGD metric) obtained by **LMOEA-DSNS** and five compared algorithms on 500-, 1000-, 2000- and 5000-D 3-objective LSMOP1-9 problems. The best result in each row is highlighted in gray.

Problem	M	D	DGEA-RVEA	LCSA	LMOCSO	WOF-NSGA-II	LMOEA-DS	LMOEA-DSNS
LSMOP1	3	500	7.1741e−1 (9.78e−2) −	6.3000e−1 (2.78e−2) −	1.3985e+0 (1.09e−1) −	5.3184e−1 (3.22e−2) −	4.5180e−1 (4.12e−2) −	4.0352e−1 (3.03e−2)
	3	1000	7.1652e−1 (1.34e−1) −	7.7427e−1 (2.77e−2) −	1.5016e+0 (8.43e−2) −	6.0355e−1 (3.58e−2) −	4.6657e−1 (3.44e−2) −	4.2747e−1 (4.73e−2)
	3	2000	7.2784e−1 (1.66e−1) −	8.4562e−1 (1.89e−2) −	1.5793e+0 (8.14e−2) −	6.3621e−1 (3.00e−2) −	5.0880e−1 (6.52e−2) −	4.2841e−1 (3.78e−2)
	3	5000	7.5406e−1 (1.02e−1) −	8.6002e−1 (1.02e−3) −	1.5626e+0 (7.73e−2) −	7.0879e−1 (3.47e−2) −	5.2793e−1 (6.95e−2) −	4.2738e−1 (4.04e−2)
LSMOP2	3	500	5.2096e−2 (2.04e−3) −	7.9226e−2 (2.42e−3) −	5.1741e−2 (5.61e−4) −	7.9191e−2 (2.65e−3) −	4.5491e−2 (1.30e−3) ≈	4.5357e−2 (1.41e−3)
	3	1000	4.1084e−2 (1.02e−3) −	6.1620e−2 (2.71e−3) −	4.0786e−2 (3.84e−4) −	5.9341e−2 (1.81e−3) −	3.8212e−2 (1.15e−3) −	3.6062e−2 (3.01e−4)
	3	2000	3.5479e−2 (3.21e−4) −	5.3130e−2 (2.32e−3) −	3.5485e−2 (1.53e−4) −	5.2758e−2 (3.20e−3) −	3.4714e−2 (1.07e−3) −	3.2056e−2 (1.50e−4)
	3	5000	3.2736e−2 (2.16e−4) −	5.0658e−2 (2.91e−3) −	3.2766e−2 (3.64e−5) −	4.9702e−2 (2.19e−3) −	3.3478e−2 (9.10e−4) −	3.0378e−2 (9.83e−5)
LSMOP3	3	500	1.1127e+0 (1.03e+0) −	8.6072e−1 (1.14e−16) −	1.3712e+1 (2.50e+0) −	1.0894e+0 (8.89e−1) −	8.5993e−1 (2.54e−3) ≈	8.5101e−1 (2.91e−2)
	3	1000	1.0575e+0 (6.05e−1) −	8.6072e−1 (1.14e−16) −	1.3265e+1 (2.07e+0) −	9.8602e−1 (4.45e−1) −	8.6192e−1 (2.81e−3) ≈	8.5692e−1 (1.56e−2)
	3	2000	1.3704e+0 (1.27e+0) −	8.6072e−1 (1.14e−16) −	1.5309e+1 (3.59e+0) −	8.8926e−1 (6.75e−2) −	8.6171e−1 (3.37e−3) ≈	8.5067e−1 (2.53e−2)
	3	5000	1.3484e+0 (7.49e−1) −	8.6072e−1 (1.14e−16) −	1.3900e+1 (1.43e+0) −	9.8415e−1 (2.65e−1) −	8.6207e−1 (2.66e−3) −	8.5926e−1 (4.56e−3)
LSMOP4	3	500	1.2354e−1 (6.67e−3) −	2.1020e−1 (3.98e−3) −	1.5170e−1 (1.30e−3) −	2.0767e−1 (3.45e−3) −	1.0552e−1 (2.13e−3) +	1.1126e−1 (4.81e−3)
	3	1000	7.7169e−2 (6.78e−3) −	1.3566e−1 (2.42e−3) −	9.2984e−2 (6.59e−4) −	1.3356e−1 (3.40e−3) −	6.8115e−2 (1.41e−3) +	7.0877e−2 (2.17e−3)
	3	2000	5.2146e−2 (2.68e−3) −	8.7461e−2 (2.54e−3) −	5.9126e−2 (2.77e−4) −	8.7003e−2 (4.50e−3) −	4.8899e−2 (7.70e−4) −	4.6943e−2 (1.06e−3)
	3	5000	3.7874e−2 (9.51e−4) −	5.9882e−2 (2.91e−3) −	3.9772e−2 (1.56e−4) −	5.7735e−2 (2.69e−3) −	3.7167e−2 (1.11e−3) −	3.4437e−2 (3.63e−4)
LSMOP5	3	500	7.3361e−1 (3.34e−1) −	5.4085e−1 (1.70e−5) −	2.9009e+0 (1.67e−1) −	5.3394e−1 (2.71e−2) −	5.2659e−1 (1.36e−2) ≈	5.2642e−1 (1.03e−2)
	3	1000	9.7423e−1 (5.67e−1) −	5.4087e−1 (3.18e−5) −	3.0758e+0 (1.08e−1) −	5.4621e−1 (2.74e−2) −	5.3235e−1 (1.03e−2) ≈	5.2872e−1 (8.40e−3)
	3	2000	7.6687e−1 (3.93e−1) −	5.4088e−1 (2.35e−5) −	3.2170e+0 (1.48e−1) −	5.4747e−1 (3.07e−2) −	5.3685e−1 (3.57e−3) −	5.2909e−1 (8.28e−3)
	3	5000	1.1304e+0 (8.42e−1) −	5.4088e−1 (2.04e−5) −	3.3200e+0 (1.06e−1) −	5.4194e−1 (2.80e−3) −	5.3891e−1 (1.25e−3) −	5.3099e−1 (7.19e−3)
LSMOP6	3	500	2.8404e+1 (4.45e+1) ≈	1.2898e+0 (1.17e−3) −	2.4955e+2 (7.56e+1) −	1.3116e+0 (4.93e−2) −	7.9523e−1 (5.36e−2) −	7.3402e−1 (2.48e−2)
	3	1000	3.8045e+1 (6.68e+1) ≈	1.3141e+0 (4.27e−4) −	3.7187e+2 (8.51e+1) −	1.3353e+0 (3.67e−2) −	8.1333e−1 (4.35e−2) −	7.5369e−1 (6.42e−2)
	3	2000	2.1062e+1 (9.08e+1) ≈	1.3248e+0 (1.55e−4) −	5.4114e+2 (1.61e+2) −	1.4292e+0 (1.62e−1) −	7.9769e−1 (4.02e−2) −	7.4151e−1 (3.11e−2)
	3	5000	2.1428e+2 (7.00e+2) ≈	1.3310e+0 (4.18e−5) −	5.2749e+2 (1.35e+2) −	1.5601e+0 (3.41e−1) −	7.6353e−1 (2.80e−2) −	7.3022e−1 (2.20e−2)
LSMOP7	3	500	9.7245e−1 (7.80e−2) −	9.1106e−1 (3.24e−3) −	1.1571e+0 (3.90e−2) −	8.8433e−1 (2.01e−2) −	8.8605e−1 (4.88e−3) −	8.7735e−1 (3.80e−3)
	3	1000	9.3124e−1 (6.39e−2) −	8.7129e−1 (6.86e−4) −	1.0500e+0 (2.66e−2) −	8.6662e−1 (2.39e−2) −	8.5880e−1 (2.03e−3) −	8.5431e−1 (1.81e−3)
	3	2000	9.1658e−1 (6.74e−2) −	8.5244e−1 (3.53e−4) −	9.9966e−1 (7.63e−3) −	8.6687e−1 (3.15e−2) −	8.4559e−1 (8.43e−4) −	8.4367e−1 (6.75e−4)
	3	5000	8.6485e−1 (6.21e−2) ≈	8.4212e−1 (2.53e−4) −	9.6666e−1 (4.23e−3) −	8.4904e−1 (2.56e−2) −	8.3497e−1 (1.82e−2) +	8.3847e−1 (2.62e−4)
LSMOP8	3	500	1.8617e−1 (3.59e−2) +	3.5776e−1 (7.02e−3) −	6.2923e−1 (9.26e−2) −	3.4771e−1 (2.28e−2) −	2.7908e−1 (4.31e−2) −	2.2576e−1 (6.92e−2)
	3	1000	2.9837e−1 (2.14e−1) ≈	3.3684e−1 (2.61e−2) −	6.3800e−1 (1.37e−1) −	2.8859e−1 (4.44e−2) −	2.4226e−1 (4.60e−2) ≈	2.1839e−1 (4.44e−2)
	3	2000	2.8759e−1 (1.73e−1) ≈	3.3120e−1 (2.64e−2) −	5.9378e−1 (8.79e−2) −	3.1175e−1 (3.96e−2) −	2.2370e−1 (3.53e−2) ≈	2.0679e−1 (4.86e−2)
	3	5000	5.0628e−1 (2.69e−1) −	3.1574e−1 (2.86e−2) −	6.0589e−1 (1.20e−1) −	2.9636e−1 (4.32e−2) −	2.2416e−1 (2.10e−2) −	1.9918e−1 (3.43e−2)
LSMOP9	3	500	2.0995e+1 (7.95e+0) −	1.5379e+0 (1.68e−4) −	2.9601e+0 (3.81e+0) −	1.1450e+0 (2.54e−4) −	5.4404e−1 (2.90e−3) +	5.4726e−1 (2.26e−3)
	3	1000	3.5924e+1 (1.32e+1) −	1.5379e+0 (4.56e−16) −	5.6852e+1 (2.85e+1) −	1.1447e+0 (2.84e−4) −	5.4103e−1 (3.07e−3) +	5.4414e−1 (2.17e−3)
	3	2000	4.8830e+1 (1.01e+1) −	1.5379e+0 (4.56e−16) −	8.4148e+1 (2.87e+1) −	1.1445e+0 (2.57e−4) −	5.4039e−1 (2.83e−3) +	5.4222e−1 (2.41e−3)
	3	5000	5.4675e+1 (6.45e+0) −	1.5379e+0 (4.56e−16) −	8.0569e+1 (3.49e+1) −	1.1797e+0 (1.08e−1) −	5.3878e−1 (2.50e−3) +	5.4070e−1 (2.60e−3)
+/ − / ≈			1/28/7	0/36/0	0/36/0	0/36/0	7/21/8	

Table 5
The statistical results (mean and standard values of IGD metric) obtained by **LMOEA-DSNS** and 5 compared algorithms on 500-, 1000-, 2000- and 5000-D 5-objective LSMOP1-9 problems. The best result in each row is highlighted in gray.

Problem	M	D	DGEA-RVEA	LCSA	LMOCSO	WOF-NSGA-II	LMOEA-DS	LMOEA-DSNS
LSMOP1	5	500	8.5571e−1 (1.43e−1) ≈	9.4187e−1 (4.72e−3) −	1.2339e+0 (3.54e−1) −	9.5426e−1 (6.28e−2) −	9.6159e−1 (8.93e−2) −	8.2577e−1 (1.15e−1)
	5	1000	8.9101e−1 (1.11e−1) ≈	9.4124e−1 (1.39e−3) −	1.5525e+0 (2.42e−1) −	9.6991e−1 (1.07e−1) −	9.4292e−1 (4.14e−2) −	8.5748e−1 (1.03e−1)
	5	2000	8.7585e−1 (1.26e−1) ≈	9.3982e−1 (1.35e−3) −	1.7570e+0 (1.42e−1) −	9.3790e−1 (1.77e−2) −	9.3382e−1 (1.70e−2) −	9.1125e−1 (1.65e−2)
	5	5000	8.9684e−1 (1.19e−1) ≈	9.3991e−1 (8.88e−4) −	1.7928e+0 (3.13e−1) −	9.3985e−1 (3.40e−2) −	9.6568e−1 (1.41e−1) −	8.4797e−1 (1.28e−1)
LSMOP2	5	500	1.7106e−1 (2.21e−3) −	1.8665e−1 (5.67e−3) −	1.7300e−1 (1.02e−3) −	1.8851e−1 (5.81e−3) −	1.3846e−1 (2.78e−3) −	1.3564e−1 (1.05e−3)
	5	1000	1.4653e−1 (7.78e−4) −	1.6131e−1 (3.95e−3) −	1.4712e−1 (5.78e−4) −	1.6308e−1 (2.69e−3) −	1.2769e−1 (2.68e−3) −	1.2535e−1 (1.00e−3)
	5	2000	1.3584e−1 (5.93e−4) −	1.5189e−1 (5.40e−3) −	1.3576e−1 (4.29e−4) −	1.5188e−1 (5.31e−3) −	1.2376e−1 (2.27e−3) ≈	1.2325e−1 (9.00e−4)
	5	5000	1.2970e−1 (4.14e−4) −	1.4465e−1 (3.52e−3) −	1.2945e−1 (4.18e−4) −	1.4690e−1 (5.21e−3) −	1.2251e−1 (2.15e−3) +	1.2349e−1 (1.03e−3)
LSMOP3	5	500	3.3327e+0 (3.13e+0) −	9.5883e−1 (0.00e+0) ≈	1.6007e+1 (6.34e+0) −	2.0177e+0 (1.27e+0) −	9.8316e−1 (3.85e−2) −	9.5882e−1 (2.41e−5)
	5	1000	4.6225e+0 (3.83e+0) −	9.5883e−1 (0.00e+0) ≈	3.1775e+1 (5.77e+1) −	1.6018e+0 (7.78e−1) −	1.0009e+0 (9.16e−2) −	9.5883e−1 (3.32e−6)
	5	2000	4.0604e+0 (3.72e+0) −	9.5883e−1 (0.00e+0) ≈	3.5824e+1 (4.31e+1) −	2.3741e+0 (1.64e+0) −	1.0058e+0 (1.06e−1) −	9.5883e−1 (0.00e+0)
	5	5000	3.0350e+0 (3.39e+0) −	9.5883e−1 (0.00e+0) ≈	3.2510e+1 (3.66e+1) −	1.8425e+0 (1.36e+0) −	1.0229e+0 (1.27e−1) −	9.5883e−1 (6.88e−7)
LSMOP4	5	500	2.7639e−1 (5.19e−3) −	3.2331e−1 (8.14e−3) −	2.9176e−1 (3.83e−3) −	3.2676e−1 (1.09e−2) −	2.7268e−1 (8.13e−3) −	2.6448e−1 (1.35e−2)
	5	1000	2.1433e−1 (2.83e−3) −	2.4477e−1 (5.01e−3) −	2.2660e−1 (2.56e−3) −	2.5209e−1 (6.64e−3) −	1.8783e−1 (2.69e−3) −	1.7983e−1 (3.52e−3)
	5	2000	1.7405e−1 (2.09e−3) −	1.9401e−1 (5.40e−3) −	1.7964e−1 (9.93e−4) −	1.9503e−1 (5.17e−3) −	1.5099e−1 (2.54e−3) −	1.4288e−1 (2.17e−3)
	5	5000	1.4513e−1 (1.01e−3) −	1.6373e−1 (5.46e−3) −	1.4640e−1 (4.98e−4) −	1.6147e−1 (5.48e−3) −	1.2863e−1 (2.81e−3) −	1.2526e−1 (1.22e−3)
LSMOP5	5	500	4.2766e−1 (1.91e−1) −	4.6753e−1 (1.71e−2) −	2.0404e+0 (3.73e−1) −	6.0439e−1 (1.85e−1) −	4.9379e−1 (1.91e−2) −	4.1064e−1 (3.90e−2)
	5	1000	6.6795e−1 (2.91e−1) −	4.6671e−1 (4.21e−2) −	2.9903e+0 (3.37e−1) −	5.8199e−1 (1.74e−1) −	4.9470e−1 (2.18e−2) −	4.1787e−1 (3.55e−2)
	5	2000	6.8647e−1 (2.43e−1) −	4.5237e−1 (1.50e−2) −	3.3412e+0 (3.47e−1) −	7.6721e−1 (5.82e−1) −	4.8984e−1 (1.83e−2) −	3.9818e−1 (3.05e−2)
	5	5000	8.1443e−1 (3.00e−1) −	4.4707e−1 (2.10e−2) −	3.4652e+0 (4.75e−1) −	7.2762e−1 (8.40e−1) −	4.9181e−1 (2.81e−2) −	3.9384e−1 (2.95e−2)
LSMOP6	5	500	1.2623e+0 (5.80e−1) ≈	1.2179e+0 (4.24e−2) −	5.4600e+1 (4.50e+1) −	2.5594e+0 (2.81e+0) −	1.2386e+0 (4.67e−2) −	1.0649e+0 (7.54e−2)
	5	1000	1.8141e+0 (9.61e−1) −	1.2633e+0 (4.80e−2) −	2.3462e+2 (1.67e+2) −	6.8382e+0 (1.44e+1) −	1.2372e+0 (1.20e−1) −	1.0508e+0 (7.81e−2)
	5	2000	2.1099e+0 (1.33e+0) −	1.2620e+0 (2.71e−2) −	3.8003e+2 (2.79e+2) −	6.9097e+0 (1.04e+1) −	1.2732e+0 (1.30e−1) −	1.0703e+0 (7.15e−2)
	5	5000	2.5708e+0 (1.31e+0) −	1.2591e+0 (2.91e−2) −	4.3840e+2 (2.58e+2) −	4.6782e+0 (5.84e+0) −	1.2188e+0 (1.28e−1) −	1.0882e+0 (3.81e−2)
LSMOP7	5	500	1.1205e+0 (9.08e−2) ≈	1.1047e+0 (1.31e−2) ≈	2.2107e+0 (2.08e−1) −	1.4698e+0 (2.45e−1) −	1.2206e+0 (8.75e−2) ≈	1.1830e+0 (1.45e−1)
	5	1000	1.0837e+0 (7.71e−2) ≈	1.0343e+0 (9.29e−3) +	1.7903e+0 (4.79e−2) −	1.2352e+0 (1.21e−1) −	1.1278e+0 (4.82e−2) ≈	1.1103e+0 (7.80e−2)
	5	2000	1.0329e+0 (6.72e−2) +	9.9936e−1 (7.25e−3) +	1.3598e+0 (1.88e−2) −	1.1786e+0 (1.23e−1) −	1.0876e+0 (3.11e−2) ≈	1.0872e+0 (8.50e−2)
	5	5000	1.0027e+0 (6.92e−2) +	9.7944e−1 (6.56e−3) +	1.1642e+0 (2.95e−2) −	1.1202e+0 (8.38e−2) ≈	1.0489e+0 (2.60e−2) +	1.0851e+0 (4.31e−2)
LSMOP8	5	500	3.8873e−1 (7.73e−2) ≈	3.8363e−1 (1.52e−2) −	9.9215e−1 (3.25e−2) −	4.3021e−1 (8.43e−2) −	4.2538e−1 (2.09e−2) −	3.6603e−1 (3.52e−2)
	5	1000	5.0444e−1 (2.06e−1) ≈	3.7729e−1 (7.38e−3) −	1.0522e+0 (3.78e−2) −	5.0051e−1 (1.78e−1) −	4.1017e−1 (1.67e−2) −	3.6331e−1 (4.05e−2)
	5	2000	6.3909e−1 (1.40e−1) −	3.8056e−1 (1.51e−2) ≈	1.0888e+0 (5.45e−2) −	5.9980e−1 (2.61e−1) ≈	4.2344e−1 (1.91e−2) ≈	5.0738e−1 (2.11e−1)
	5	5000	7.2486e−1 (1.04e−1) −	3.7253e−1 (8.39e−3) ≈	1.0565e+0 (1.07e−1) −	5.2629e−1 (2.30e−1) ≈	4.4629e−1 (4.95e−2) ≈	4.9766e−1 (2.02e−1)
LSMOP9	5	500	2.1930e+1 (3.02e+0) −	3.0005e+0 (1.37e−15) −	3.4203e+0 (9.69e−1) −	1.8283e+0 (1.71e−1) −	4.3396e−1 (7.84e−2) −	3.2842e−1 (2.75e−2)
	5	1000	6.3529e+1 (7.79e+0) −	3.0005e+0 (1.37e−15) −	1.6842e+2 (8.64e+1) −	1.8154e+0 (1.59e−1) −	4.2122e−1 (5.96e−2) −	3.1839e−1 (3.22e−2)
	5	2000	1.0802e+2 (1.34e+1) −	3.0005e+0 (1.37e−15) −	3.4067e+2 (7.21e+1) −	1.8602e+0 (4.19e−2) −	3.9171e−1 (5.38e−2) −	3.1709e−1 (3.64e−2)
	5	5000	1.4284e+2 (1.07e+1) −	3.0005e+0 (1.37e−15) −	3.2866e+2 (6.54e+1) −	2.0771e+0 (1.99e−1) −	3.9562e−1 (6.15e−2) −	3.1432e−1 (1.92e−2)
+/ − / ≈			2/25/9	3/26/7	0/36/0	0/33/3	2/28/6	

2, 3, 4, 5, 8 and 9. This indicates the robust performance of LMOEA-DSNS. For LSMOP6, the performance of LMOEA-DSNS is worse than the compared MOEAs. This is presumably due to the highly complex landscape of LSMOP6 in the high-dimensional objective space, which may mislead the algorithm to local optima at a late stage of evolution.

Finally, the comparative results on 7-, 9-, and 12-objective LSMOPs are summarized in Table 6. In Table 6, the proposed LMOEA-DSNS obtained overall better results on 7-, 9-, and 12-objective LSMOPs than DGEA-REVA, WOF-NSGA-II, LMOCSO, LCSA, and LMOEA-DS. Especially for the 12-objective LSMOPs, LMOEA-DSNS outperforms

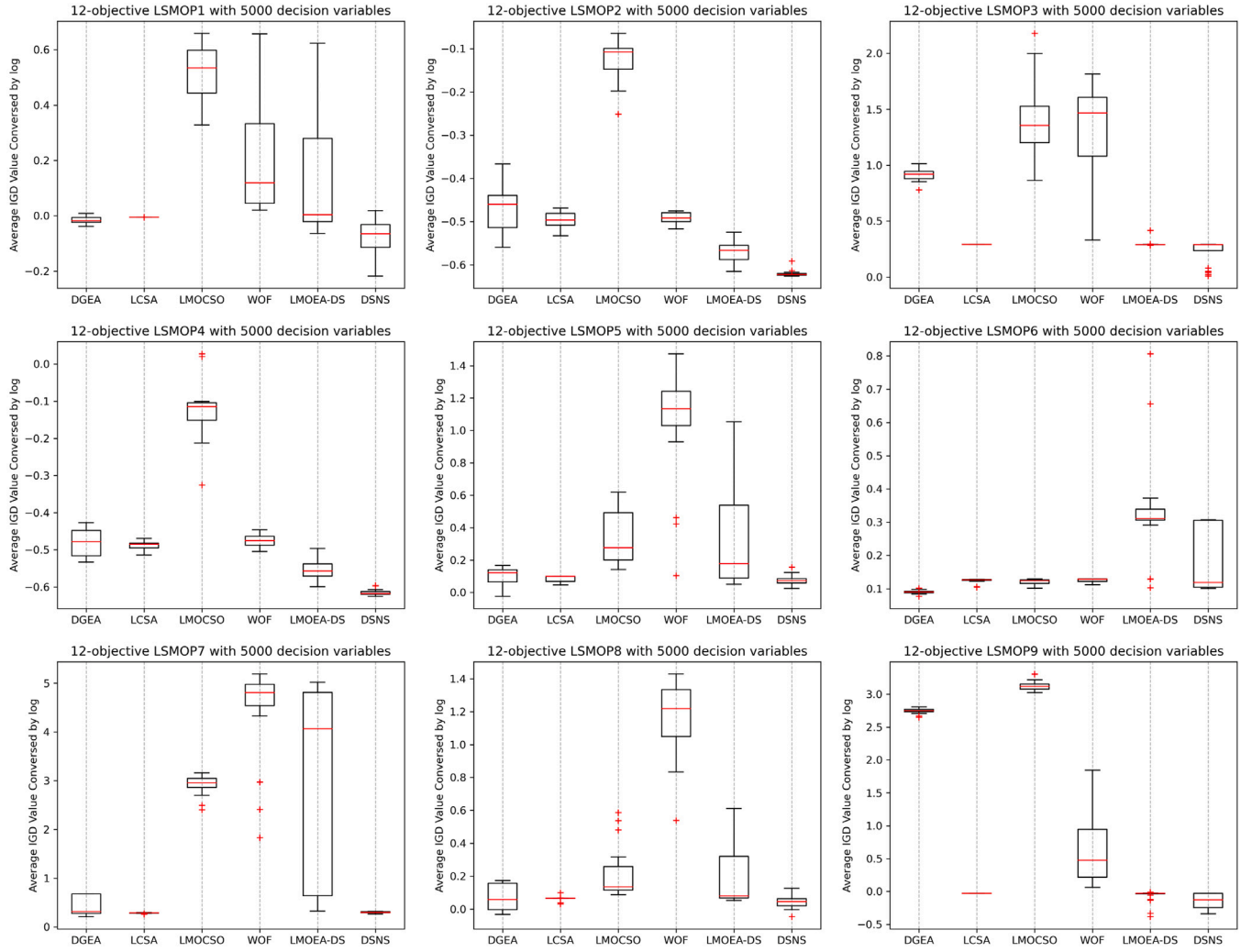

Fig. 3. The boxplots of LMOEA-DSNS and other five algorithms on 12-objective LSMOP1-9 with 5000 decision variables.

Table 6
The summary of the statistical results obtained by **LMOEA-DSNS** and 5 compared algorithms on 500-, 1000-, 2000- and 5000-D 7-, 9- and 12-objective LSMOP1-9 problems, where **LMOEA-DSNS** is better than (+), worst than (−) and comparable to (≈) each of 5 compared algorithms according to the Wilcoxon Rank sum test with bonferroni correction.

	LMOEA-DSNS	DGEA-RVEA	LCSA	LMOCSO	WOF-NSGA-II	LMOEA-DS
7	+	14/36	16/36	26/36	36/36	34/36
	≈	9/36	5/36	6/36	0/36	2/36
	−	13/36	15/36	4/36	0/36	0/36
9	+	16/36	17/36	29/36	33/36	25/36
	≈	5/36	11/36	3/36	1/36	10/36
	−	15/36	8/36	4/36	2/36	1/36
12	+	21/36	27/36	32/36	33/36	36/36
	≈	8/36	3/36	2/36	2/36	0/36
	−	7/36	6/36	2/36	2/36	0/36

DGEA-REVA, WOF-NSGA-II, LMOCSO, LCSA, and LMOEA-DS on 21/36, 33/36, 32/36, 27/36, and 36/36, respectively. In summary, these results indicate that the proposed LMOEA-DSNS has good competitive performance and is effective in handling large-scale many-objective optimization problems.

5. Conclusion

In this article, we have proposed a bi-level offspring generation architecture and a deep sampling method for large-scale multi\many-objective optimization. In the proposed architecture, the offspring generation process is divided into two phases, where the first phase generates offspring via general genetic operators to preserve convergence, and the second phase uses the proposed sampling method DSNS to generate offspring with good diversity. The existing MOEAs can be extended to our proposed bi-level architecture. On this basis, a new large-scale multi-objective algorithm, LMOEA-DSNS, was proposed.

The experimental results demonstrated that our proposed bi-level offspring regeneration architecture can improve the performance of existing MOEAs dramatically in comparison with their original versions. Moreover, the proposed LMOEA-DSNS also showed advantages over five advanced large-scale MOEAs on 3- to 12-objective large-scale multi\many-objective problems with up to 5000 dimensions.

However, as observed from the experimental results given in Tables 8, 9, and 10 of the Supplementary, the performance of the proposed DSNS was diminished in solving LSMOPs with mixed and convex landscapes (i.e., LSMOP6 and LSMOP8) in a high-dimensional objective space. For this issue, we will study how to adjust the sampling range adaptively, to better track complex PS landscapes by producing fewer invalid solutions.

In addition, the solution-sorting methods of LMOEA-DSNS will be further studied by considering a new fractional dominance method [45] and the voting mechanism [46] to strengthen convergence. Finally, it is of considerable interest to test the adaptability and scalability of the performance of the proposed algorithm in different large-scale test problems and real-world problems in the future.

CRediT authorship contribution statement

Wei Liu: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing – original draft, Resources. **Li Chen:** Supervision, Project administration. **Xingxing Hao:** Writing – review & editing. **Wei Zhou:** Data curation. **Xin Cao:** Investigation. **Fei Xie:** Visualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.swevo.2022.101152.

References

[1] A. Zhou, B. Qu, H. Li, S. Zhao, P.N. Suganthan, Q. Zhang, Multiobjective evolutionary algorithms: A survey of the state of the art, Swarm Evol. Comput. 1 (1) (2011) 32–49, http://dx.doi.org/10.1016/j.swevo.2011.03.001.

[2] M. Gong, J. Liu, H. Li, Q. Cai, L. Su, A multiobjective sparse feature learning model for deep neural networks, IEEE Trans. Neural Netw. Learn. Syst. 26 (12) (2015) 3263–3277, http://dx.doi.org/10.1109/TNNLS.2015.2469673.

[3] J. Cheney, The application of optimisation methods to the design of large scale telecommunication networks, in: IEE Colloquium on Large-Scale and Hierarchical Systems, 1988, pp. 2/1–2/2.

[4] L. Parsons, E. Haque, H. Liu, Subspace clustering for high dimensional data: A review, SIGKDD Explor. Newsl. 6 (1) (2004) 90–105, http://dx.doi.org/10.1145/1007730.1007731.

[5] H. Wang, L. Jiao, R. Shang, S. He, F. Liu, A memetic optimization strategy based on dimension reduction in decision space, Evol. Comput. 23 (1) (2015) 69–100, http://dx.doi.org/10.1162/EVCO_a_00122.

[6] L.M. Antonio, C.A.C. Coello, Use of cooperative coevolution for solving large scale multiobjective optimization problems, in: 2013 IEEE Congress on Evolutionary Computation, 2013, pp. 2758–2765, http://dx.doi.org/10.1109/CEC.2013.6557903.

[7] L. Miguel Antonio, C.A. Coello Coello, Decomposition-based approach for solving large scale multi-objective problems, in: J. Handl, E. Hart, P.R. Lewis, M. López-Ibáñez, G. Ochoa, B. Paechter (Eds.), Parallel Problem Solving from Nature – PPSN XIV, Springer International Publishing, Cham, 2016, pp. 525–534.

[8] M. Li, J. Wei, A cooperative co-evolutionary algorithm for large-scale multi-objective optimization problems, in: H.E. Aguirre, K. Takadama (Eds.), Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO 2018, July 15-19, 2018, ACM, Kyoto, Japan, 2018, pp. 1716–1721, http://dx.doi.org/10.1145/3205651.3208250.

[9] F. Sander, H. Zille, S. Mostaghim, Transfer strategies from single- to multi-objective grouping mechanisms, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 729–736, http://dx.doi.org/10.1145/3205455.3205491.

[10] A. Song, Q. Yang, W. Chen, J. Zhang, A random-based dynamic grouping strategy for large scale multi-objective optimization, in: IEEE Congress on Evolutionary Computation, CEC 2016, July 24-29, 2016, IEEE, Vancouver, BC, Canada, 2016, pp. 468–475, http://dx.doi.org/10.1109/CEC.2016.7743831.

[11] L.M. Antonio, C.A.C. Coello, S.B. González-Brambila, J.F. González, M.G.C. Tapia, Operational decomposition for large scale multi-objective optimization problems, in: M. López-Ibáñez, A. Auger, T. Stützle (Eds.), Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO 2019, July 13-17, 2019, ACM, Prague, Czech Republic, 2019, pp. 225–226, http://dx.doi.org/10.1145/3319619.3322068.

[12] B. Cao, J. Zhao, Y. Gu, Y. Ling, X. Ma, Applying graph-based differential grouping for multiobjective large-scale optimization, Swarm Evol. Comput. 53 (2020) 100626, http://dx.doi.org/10.1016/j.swevo.2019.100626.

[13] S. Liu, Q. Lin, K. Wong, L. Ma, C.A.C. Coello, D. Gong, A novel multi-objective evolutionary algorithm with dynamic decomposition strategy, Swarm Evol. Comput. 48 (2019) 182–200, http://dx.doi.org/10.1016/j.swevo.2019.02.010.

[14] 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 Trans. Evol. Comput. 20 (2) (2016) 275–298, http://dx.doi.org/10.1109/TEVC.2015.2455812.

[15] X. Zhang, Y. Tian, R. Cheng, Y. Jin, A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization, IEEE Trans. Evol. Comput. 22 (1) (2018) 97–112, http://dx.doi.org/10.1109/TEVC.2016.2600642.

[16] R. Liu, R. Ren, J. Liu, J. Liu, A clustering and dimensionality reduction based evolutionary algorithm for large-scale multi-objective problems, Appl. Soft Comput. 89 (2020) 106120, http://dx.doi.org/10.1016/j.asoc.2020.106120.

[17] S. Liu, Q. Lin, Y. Tian, K.C. Tan, A variable importance-based differential evolution for large-scale multiobjective optimization, IEEE Trans. Cybern. (2021) 1–15, http://dx.doi.org/10.1109/TCYB.2021.3098186.

[18] H. Chen, X. Zhu, W. Pedrycz, S. Yin, G. Wu, H. Yan, PEA: parallel evolutionary algorithm by separating convergence and diversity for large-scale multi-objective optimization, in: 38th IEEE International Conference on Distributed Computing Systems, ICDCS 2018, July 2-6, 2018, IEEE Computer Society, Vienna, Austria, 2018, pp. 223–232, http://dx.doi.org/10.1109/ICDCS.2018.00031.

[19] W. Du, L. Tong, Y. Tang, A framework for high-dimensional robust evolutionary multi-objective optimization, in: H.E. Aguirre, K. Takadama (Eds.), Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO 2018, July 15-19, 2018, ACM, Kyoto, Japan, 2018, pp. 1791–1796, http://dx.doi.org/10.1145/3205651.3208243.

[20] W. Du, W. Zhong, Y. Tang, W. Du, Y. Jin, High-dimensional robust multi-objective optimization for order scheduling: A decision variable classification approach, IEEE Trans. Ind. Informatics 15 (1) (2019) 293–304, http://dx.doi.org/10.1109/TII.2018.2836189.

[21] H. Zille, H. Ishibuchi, S. Mostaghim, Y. Nojima, A framework for large-scale multiobjective optimization based on problem transformation, IEEE Trans. Evol. Comput. 22 (2) (2018) 260–275, http://dx.doi.org/10.1109/TEVC.2017.2704782.

[22] C. He, L. Li, Y. Tian, X. Zhang, R. Cheng, Y. Jin, X. Yao, Accelerating large-scale multiobjective optimization via problem reformulation, IEEE Trans. Evol. Comput. 23 (6) (2019) 949–961, http://dx.doi.org/10.1109/TEVC.2019.2896002.

[23] R. Liu, J. Liu, Y. Li, J. Liu, A random dynamic grouping based weight optimization framework for large-scale multi-objective optimization problems, Swarm Evol. Comput. 55 (2020) 100684, http://dx.doi.org/10.1016/j.swevo.2020.100684.

[24] C. He, R. Cheng, Y. Tian, X. Zhang, Iterated problem reformulation for evolutionary large-scale multiobjective optimization, in: IEEE Congress on Evolutionary Computation, CEC 2020, July 19-24, 2020, IEEE, Glasgow, United Kingdom, 2020, pp. 1–8, http://dx.doi.org/10.1109/CEC48606.2020.9185553.

[25] S. Qin, C. Sun, Y. Jin, Y. Tan, J. Fieldsend, Large-scale evolutionary multi-objective optimization assisted by directed sampling, IEEE Trans. Evol. Comput. (2021) 1, http://dx.doi.org/10.1109/TEVC.2021.3063606.

[26] Y. Tian, X. Zheng, X. Zhang, Y. Jin, Efficient large-scale multiobjective optimization based on a competitive swarm optimizer, IEEE Trans. Cybern. 50 (8) (2020) 3696–3708, http://dx.doi.org/10.1109/TCYB.2019.2906383.

[27] Z. Ding, L. Chen, D. Sun, X. Zhang, A multi-stage knowledge-guided evolutionary algorithm for large-scale sparse multi-objective optimization problems, Swarm Evol. Comput. 73 (2022) 101119, http://dx.doi.org/10.1016/j.swevo.2022.101119.

[28] C. He, R. Cheng, D. Yazdani, Adaptive offspring generation for evolutionary large-scale multiobjective optimization, IEEE Trans. Syst., Man, Cybern: Syst. (2020) 1–13, http://dx.doi.org/10.1109/TSMC.2020.3003926.

[29] C. He, S. Huang, R. Cheng, K.C. Tan, Y. Jin, Evolutionary multiobjective optimization driven by generative adversarial networks (GANs), IEEE Trans. Cybern. 51 (6) (2021) 3129–3142, http://dx.doi.org/10.1109/TCYB.2020.2985081.

[30] Z. Liang, Y. Li, Z. Wan, Large scale many-objective optimization driven by distributional adversarial networks, 2020, CoRR abs/2003.07013. arXiv:2003.07013. URL https://arxiv.org/abs/2003.07013.

[31] Y. Tian, L. Si, X. Zhang, R. Cheng, C. He, K. Tan, Y. Jin, Evolutionary large-scale multi-objective optimization: A survey, ACM Comput. Surv. (2021).

[32] C. Hillermeier, et al., Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach, Vol. 135, Springer Science & Business Media, 2001.

[33] S. Sayin, Kaisa M. Miettinen nonlinear multiobjective optimization Kluwer academic publishers, 1999 ISBN 0-7923-8278-1, 320 pages, European J. Oper. Res. 148 (1) (2003) 229–230, http://dx.doi.org/10.1016/S0377-2217(02)00303-X.

[34] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, Test problems for large-scale multiobjective and many-objective optimization, IEEE Trans. Cybern. 47 (12) (2017) 4108–4121, http://dx.doi.org/10.1109/TCYB.2016.2600577.

[35] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197, http://dx.doi.org/10.1109/4235.996017.

[36] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 20 (5) (2016) 773–791, http://dx.doi.org/10.1109/TEVC.2016.2519378.

[37] S. Kukkonen, J. Lampinen, GDE3: the third evolution step of generalized differential evolution, in: 2005 IEEE Congress on Evolutionary Computation, Vol. 1, 2005, pp. 443–450, http://dx.doi.org/10.1109/CEC.2005.1554717, Vol.1.

[38] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, E.P.K. Tsang, Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion, in: IEEE International Conference on Evolutionary Computation, CEC 2006, Part of WCCI 2006, 16-21 July 2006, IEEE, Vancouver, BC, Canada, 2006, pp. 892–899, http://dx.doi.org/10.1109/CEC.2006.1688406.

[39] H. Ishibuchi, R. Imada, Y. Setoguchi, Y. Nojima, How to specify a reference point in hypervolume calculation for fair performance comparison, Evol. Comput. 26 (3) (2018) http://dx.doi.org/10.1162/evco_a_00226.

[40] P. Czyżżak, A. Jaszkiewicz, Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization, J. Multi-Crit. Decis. Anal. 7 (1) (1998) 34–47.

[41] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18, http://dx.doi.org/10.1016/j.swevo.2011.02.002.

[42] H. Zille, Large-scale multi-objective optimisation: New approaches and a classification of the state-of-the-art, 2019.

[43] Y. Tian, R. Cheng, X. Zhang, Y. Jin, PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum], IEEE Comput. Intell. Mag. 12 (4) (2017) 73–87, http://dx.doi.org/10.1109/MCI.2017.2742868.

[44] H. Ishibuchi, L.M. Pang, K. Shang, Difficulties in fair performance comparison of multi-objective evolutionary algorithms [research frontier], IEEE Comput. Intell. Mag. 17 (1) (2022) 86–101, http://dx.doi.org/10.1109/MCI.2021.3129961.

[45] W. Qiu, J. Zhu, G. Wu, M. Fan, P.N. Suganthan, Evolutionary many-Objective algorithm based on fractional dominance relation and improved objective space decomposition strategy, Swarm Evol. Comput. 60 (2021) 100776, http://dx.doi.org/10.1016/j.swevo.2020.100776.

[46] W. Qiu, J. Zhu, G. Wu, H. Chen, W. Pedrycz, P.N. Suganthan, Ensemble many-objective optimization algorithm based on voting mechanism, IEEE Trans. Syst. Man Cybern. Syst. 52 (3) (2022) 1716–1730, http://dx.doi.org/10.1109/TSMC.2020.3034180.